Detecting community structure in networks

M.E.J. Newman's results[1,2] (presented by Botond Szabo)

[1]Detecting community structure in networks (2004)

[2]Finding community structure in networks using eigenvectors of matrices (2006)

Statistics for Structures Seminar
Amsterdam, 01. 04. 2015.

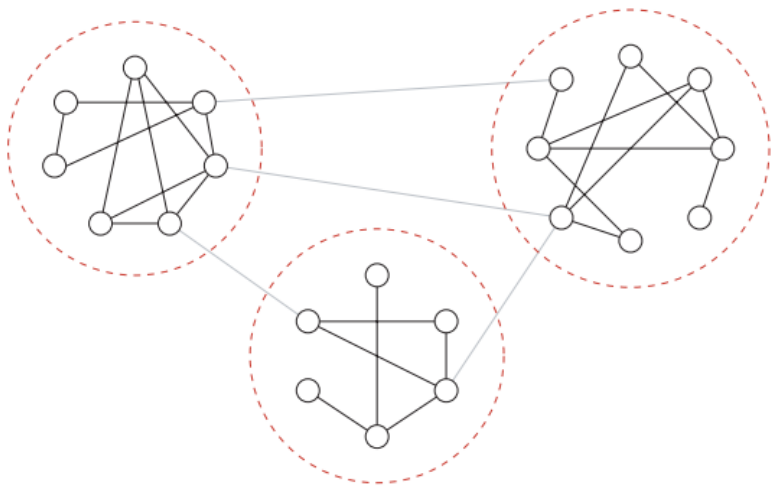
Outline

- Introduction
- Bisection Algorithms
  - Spectral algorithm (Laplacian)
  - The Kernighan-Lin algorithm (greedy)
  - Modularity algorithm
- Multisection Algorithms
  - Girvan and Newman algorithm
  - Generalized modularity algorithm
- Conclusion

Model

Model: Grap $G = (V, E)$, with unweighted vertices V and undirected, unweighted edges E.

Goal: Find communities:



Examples: Social networks, biochemical networks, information networks (parallel computing)

Spectral algorithm I.

Definition: Laplacian $L = D - A$,
where D is the diagonal matrix of vertex degrees and A is the adjacency matrix.

Properties:

- Since $D_{i,i} = \sum_j A_{i,j}$ the vector $\mathbf{v}_1 = (1, 1, .., 1)$ is an eigenvector of L with $\lambda_1 = 0$ eigenvalue.
- All eigenvalues λ_i are non-negative.
- The # of zero eigenvalues gives the # of components.
- In symmetric matrices the eigenvectors corresponding to different eigenvalues are orthogonal.
- In connected graphs the eigenvectors contain both positive and negative components (except $\mathbf{v}_1$).

Spectral algorithm II.

Application: Consider the problem of finding two communities in a connected graph.

Goal: Minimize the cut size

$$R = \frac{1}{2} \sum_{\substack{i,j \text{ in diffe-} \\ \text{rent groups}}} A_{i,j} = \frac{1}{4} \mathbf{s}^T \mathbf{L}\, \mathbf{s} = \sum_{i=1}^{n} a_i^2 \lambda_i,$$

where $s_i = \pm 1$ (group indicator), $\mathbf{s} = \sum_{i=1}^{n} a_i \mathbf{v}_i$.

Spectral algorithm II.

Application: Consider the problem of finding two communities in a connected graph.

Goal: Minimize the cut size

$$R = \frac{1}{2} \sum_{\substack{i,j \text{ in diffe-} \\ \text{rent groups}}} A_{i,j} = \frac{1}{4} \mathbf{s}^T \mathbf{L}\, \mathbf{s} = \sum_{i=1}^{n} a_i^2 \lambda_i,$$

where $s_i = \pm 1$ (group indicator), $\mathbf{s} = \sum_{i=1}^{n} a_i \mathbf{v}_i$.

Problem: The minimum of R is taken in the trivial case $s = (1, 1, ..., 1)$.

Spectral algorithm III.

Solution:

- Fix the size of the two groups (n_1, n_2). Then

$$a_1^2 = (\mathbf{v}_1^T \mathbf{s})^2 = (n_1 - n_2)^2 / n.$$

- Ideally $\mathbf{s}$ proportional to $\mathbf{v}_2$, but $s_i \in \{-1, 1\}$.
- Choose $\mathbf{s}$ close to proportional to $\mathbf{v}_2$:

$$s_i = \begin{cases} +1 & \text{if } v_i^{(2)} \geq 0, \\ -1 & \text{if } v_i^{(2)} < 0. \end{cases} \tag{1}$$

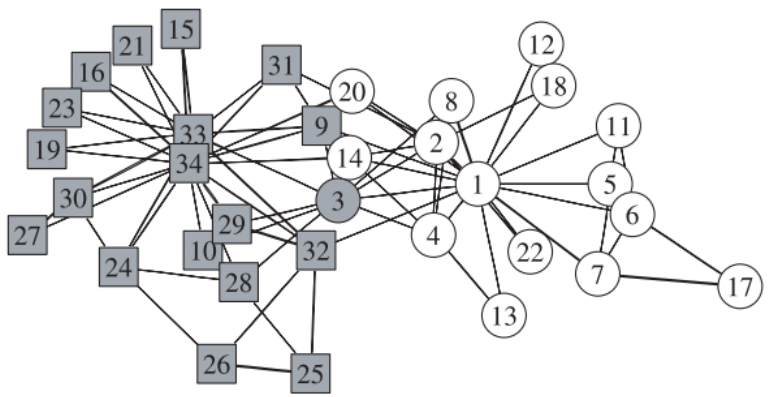
- If $\#\{v_i^{(2)} \geq 0\} > n_1$, then assign the smallest one to the other group.

Alternative spectral algorithm

Approximate algorithm: No size control on communities, using ideas from above:

$$s_i = \begin{cases} +1 & \text{if } v_i^{(2)} \geq 0, \\ -1 & \text{if } v_i^{(2)} < 0. \end{cases} \tag{2}$$

Example: The karate club



(a)

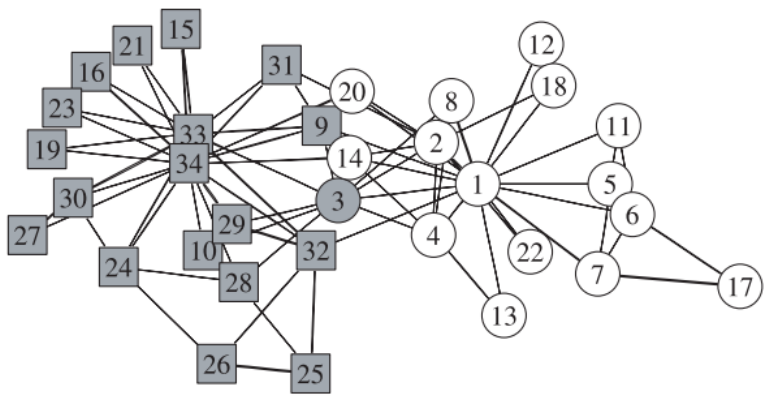
Runtime: $O(n^3)$, for sparse Laplacian $m/(\lambda_3 - \lambda_2)$.

Alternative spectral algorithm

Approximate algorithm: No size control on communities, using ideas from above:

$$s_i = \begin{cases} +1 & \text{if } v_i^{(2)} \geq 0, \\ -1 & \text{if } v_i^{(2)} < 0. \end{cases} \tag{2}$$

Example: The karate club



(a)

Runtime: $O(n^3)$, for sparse Laplacian $m/(\lambda_3 - \lambda_2)$.
Alternatively: Minimize the ratio cut $R/(n_1 n_2)$, instead of R.

Discussion of Spectral algorithms

Problem: Satisfactory if the network does not divide up easily into groups but one has to do the best. However, they don't reflect our intuitively concept of network communities.

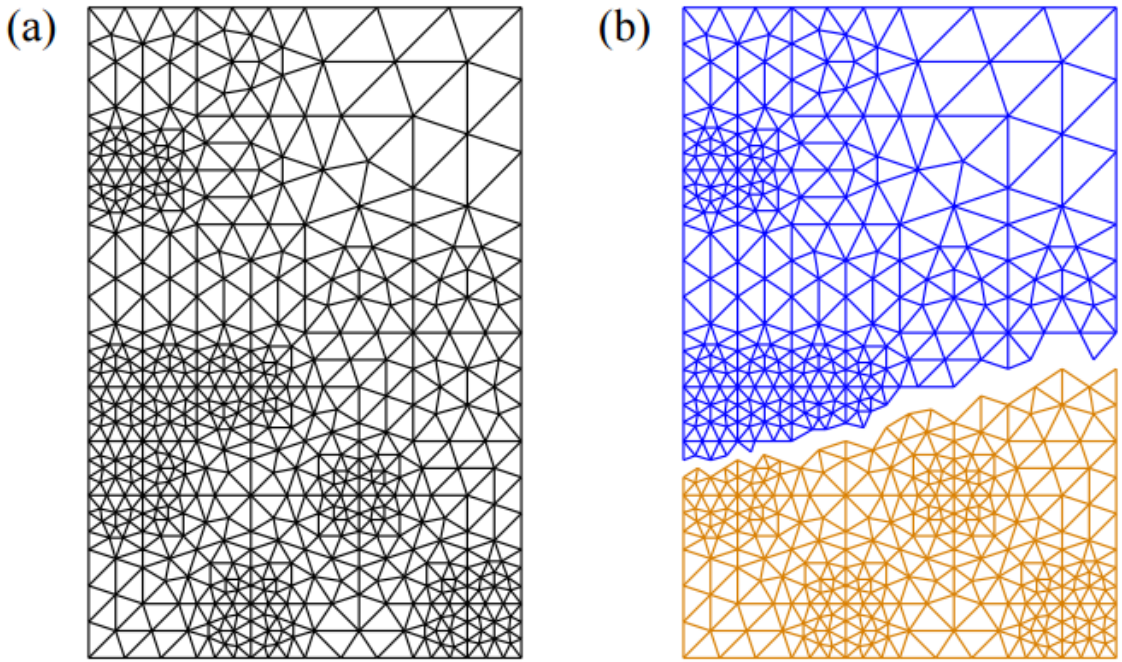


FIG. 1 (a) The mesh network of Bern *et al.* [49]. (b) The best division into equal-sized parts found by the spectral partitioning algorithm based on the Laplacian matrix.

Kernighan-Lin algorithm

Algorithm:

- Assume that we know the community sizes $|G_1|, |G_2|$
- Assign benefit function for every division:
  Q= # edges within − # edges between the two groups.
- Stage 1: Maximize ΔQ over all pairs $i \in G_1$, $j \in G_2$.
- Then switch vertices and repeat until from one group all vertices have been swapped.
- Stage 2: Choose in the preceding sequence the maximum Q.

Runtime: worst case $O(n^2)$.

Example: Perfect match in the karate club.

Modularity

Problem:

- We usually don't know the size of the communities.
- The number of edges between communities is smaller than expected.

Modularity

Problem:

- We usually don't know the size of the communities.
- The number of edges between communities is smaller than expected.

Definition: modularity - Benefit function (different, but related to before):
$Q = \#$ edges within communities - expected $\#$ of such edges.

Second term is rather vague. What do we mean under it?

Modularity

Problem:

- We usually don't know the size of the communities.
- The number of edges between communities is smaller than expected.

Definition: modularity - Benefit function (different, but related to before):
$Q = \#$ edges within communities - expected $\#$ of such edges.

Second term is rather vague. What do we mean under it?

Null model: n vertices, $P_{i,j}$ the probability of an edge between i and j. Then

$$Q = \frac{1}{2m} \sum_{i,j} [A_{i,j} - P_{i,j}] \delta(g_i, g_j),$$

where g_i denotes the community i belongs to.

Choice of $P_{i,j}$

Condition 1:

$$\sum_{i,j} P_{i,j} = \sum_{i,j} A_{i,j} = 2m.$$

Example: Bernoulli model $P_{i,j} = p$, which has binomial degree distribution, not right skewed like most of real-world networks.

Choice of $P_{i,j}$

Condition 1:

$$\sum_{i,j} P_{i,j} = \sum_{i,j} A_{i,j} = 2m.$$

Example: Bernoulli model $P_{i,j} = p$, which has binomial degree distribution, not right skewed like most of real-world networks.

Condition 2:

$$\sum_{j} P_{i,j} = \sum_{j} A_{i,j} =: k_i$$

which for entirely random edges leads to

$$P_{i,j} = \frac{k_i k_j}{2m}.$$

This is closely related to the configuration model (preferal attachment).

Spectral optimization of modularity

Assumption: we have two communities, but no fixed size.

Definition: Modularity matrix

- Rewrite modularity function

$$Q = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s} = \frac{1}{4m}\sum_i a_i^2\beta_i,$$

  where $\mathbf{B}$=$\mathbf{A}$-$\mathbf{P}$ and $\mathbf{s} = \sum_{i=1}^{n} a_i\mathbf{u}_i$ (β_i is the eigenvalue corresponding to the eigenvector $\mathbf{u}_i$ of $\mathbf{B}$)
- There exists i, such that $\beta_i = 0$ and $\mathbf{v}_i = (1, 1, ..., 1)$.
- But there could be (and in practice are) both positive and negative eigenvalues.

Spectral optimization of modularity II

Solution: similarly to the spectral algorithm

- Best would be to have $\mathbf{s}$ proportional to $\mathbf{u}_1$ (with largest β_1).
- But $s_i = \pm 1$.
- Therefore take

$$s_i = \begin{cases} +1 & \text{if } u_i^{(1)} \geq 0, \\ -1 & \text{if } u_i^{(1)} < 0. \end{cases} \tag{3}$$

Runtime: $O(n^2)$ (by using Lanczos method or its variants).

Example: Modularity

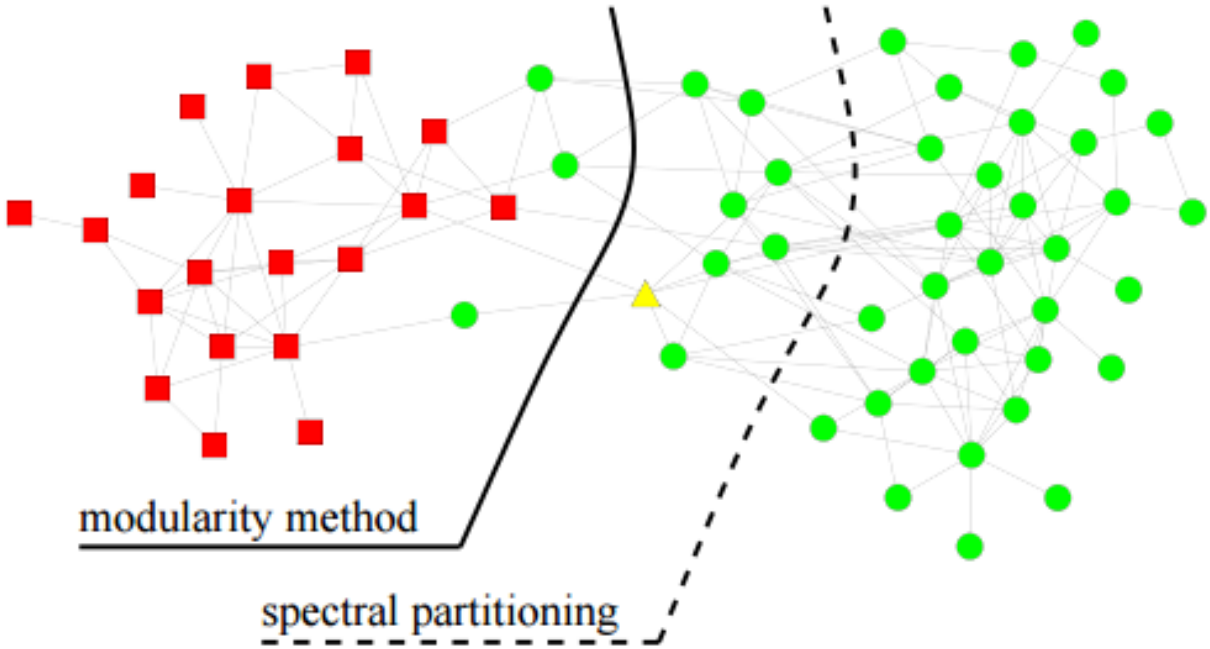


FIG. 2 The dolphin social network of Lusseau *et al.* [68]. The dashed curve represents the division into two equally sized parts found by a standard spectral partitioning calculation (Section II). The solid curve represents the division found by the modularity-based method of this section. And the squares and circles represent the actual division of the network observed when the dolphin community split into two as a result of the departure of a keystone individual. (The individual who departed is represented by the triangle.)

Negative Eigenvalues

Question: what information are stored in the negative eigenvalues?

Negative Eigenvalues

Question: what information are stored in the negative eigenvalues?

Answer: "Anti-community structure", i.e. numbers of edges within groups are smaller than expected.

Procedure:

- Minimize modularity: take $\mathbf{s}$ almost parallel to $\mathbf{v}_n$ (corresponding β_n).

$$s_i = \begin{cases} +1 & \text{if } u_i^{(n)} \geq 0, \\ -1 & \text{if } u_i^{(n)} < 0. \end{cases} \tag{4}$$

- Refinement step: move single vertices between groups to minimize modularity.

Negative Eigenvalues

Question: what information are stored in the negative eigenvalues?

Answer: "Anti-community structure", i.e. numbers of edges within groups are smaller than expected.

Procedure:

- Minimize modularity: take $\mathbf{s}$ almost parallel to $\mathbf{v}_n$ (corresponding β_n).

$$s_i = \begin{cases} +1 & \text{if } u_i^{(n)} \geq 0, \\ -1 & \text{if } u_i^{(n)} < 0. \end{cases} \tag{4}$$

- Refinement step: move single vertices between groups to minimize modularity.

Other uses:

- Network correlation: Adjacency vertices have similar properties.
- Community centrality: How central vertices are in their community.

Example: Anti-community structure

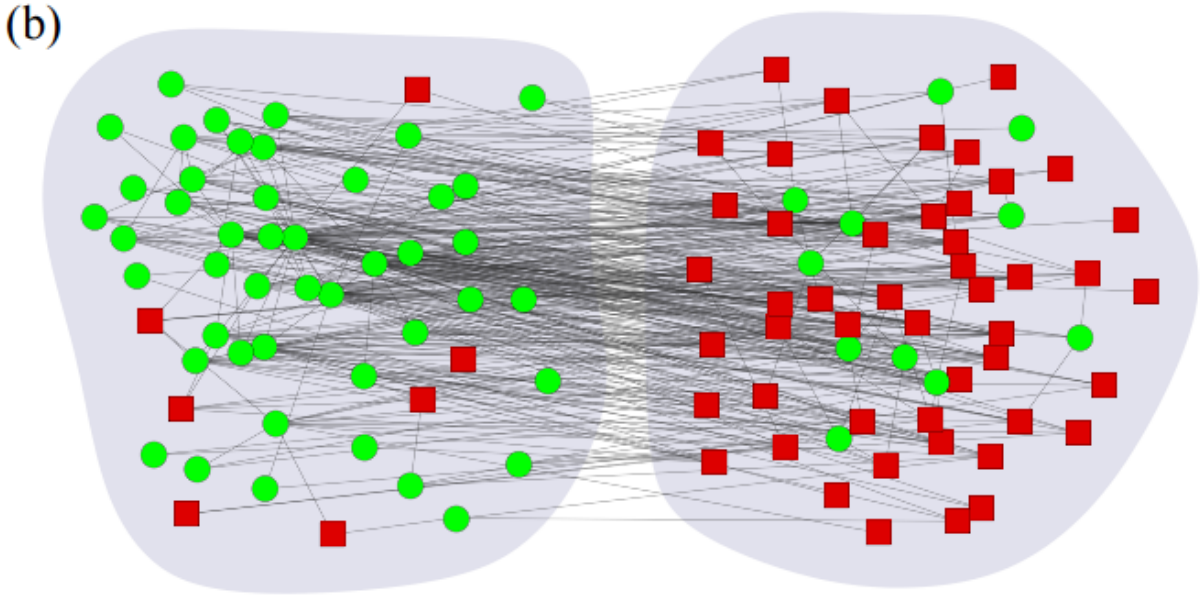
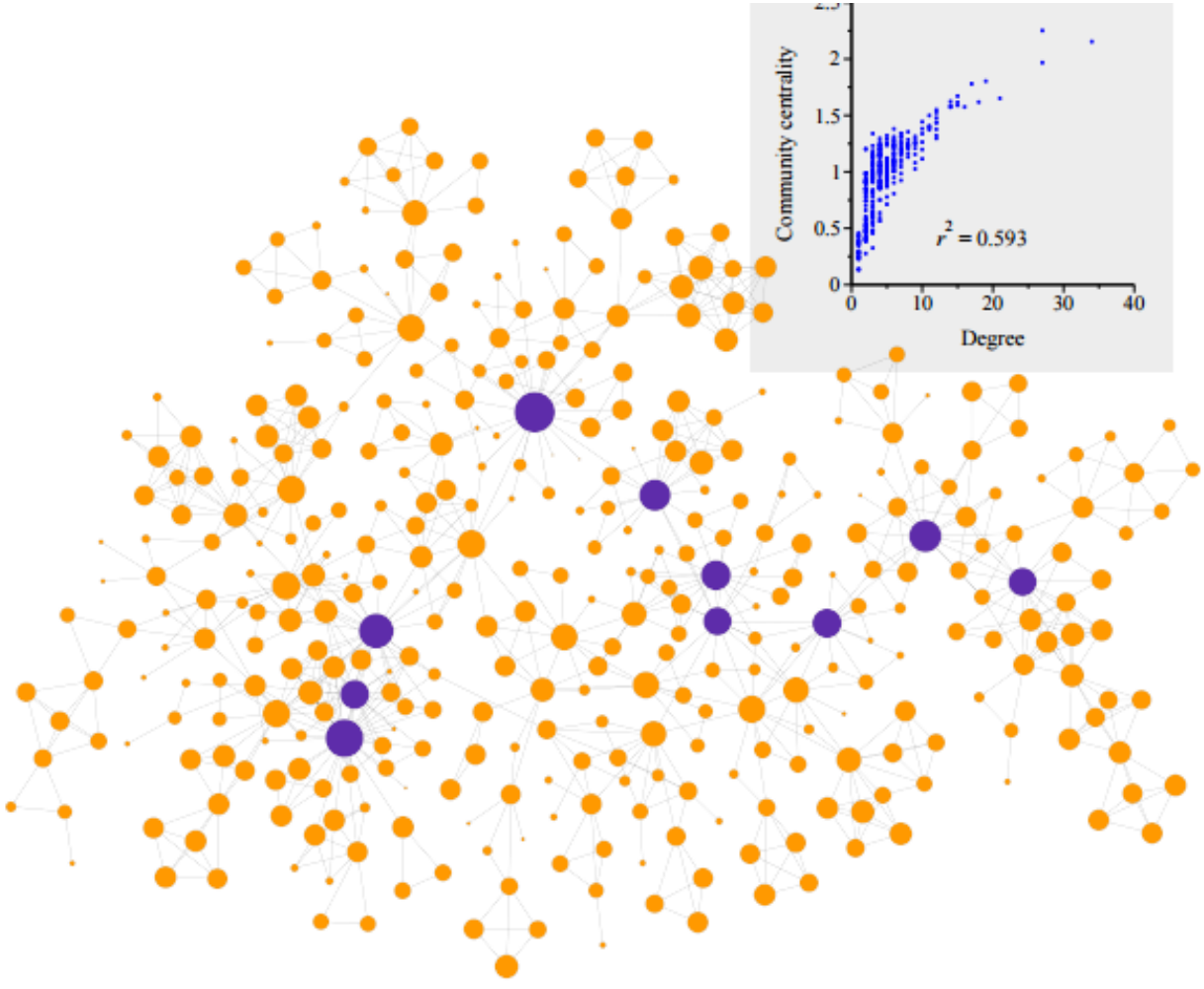


FIG. 7 (a) The network of commonly occurring English adjectives (circles) and nouns (squares) described in the text. (b) The same network redrawn with the nodes grouped so as to minimize the modularity of the grouping. The network is now revealed to be approximately bipartite, with one group consisting almost entirely of adjectives and the other of nouns.

Example: Community centrality



IG. 8 A network of coauthorships between 379 scientists whose research centers on the properties of networks of one kind or nother. Vertex diameters indicate the community centrality and the ten vertices with highest centralities are highlighted. For hose readers curious about the identities of the vertices, an annotated version of this figure, names and all, can be found at `ttp://www.umich.edu/~mejn/centrality`. Inset: a scatter plot of community centrality against vertex degrees. Like most entrality measures, this one is correlated with degree, though only moderately strongly.

Multiple communities

Problem: In many real-world examples we don't know the numbers of the communities.

Multiple communities

Problem: In many real-world examples we don't know the numbers of the communities.
Approach: Repeated division into two: not ideal.

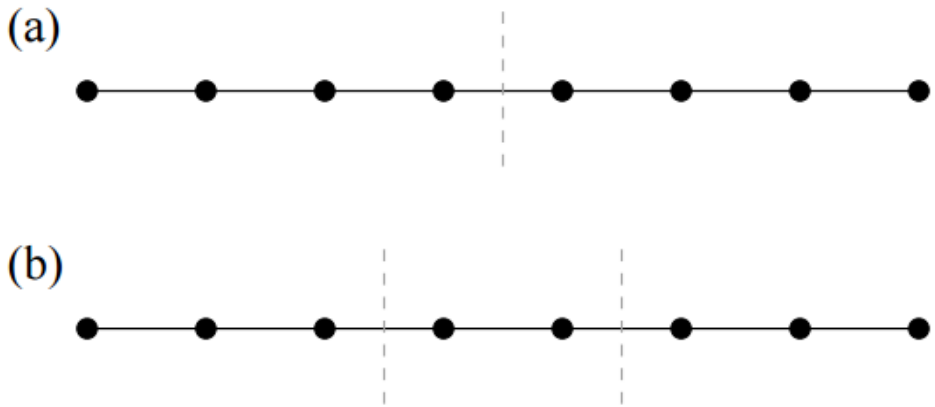


FIG. 5 Division by the method of optimal modularity of a simple network consisting of eight vertices in a line. (a) The optimal division into just two parts separates the network symmetrically into two groups of four vertices each. (b) The optimal division into any number of parts divides the network into three groups as shown here.

Girvan and Newman algorithm

Idea: Remove edges from the networks, with high "betweenness score", iteratively.

Motivation: Few edges between communities are bottlenecks. Traffic has to travel through them.

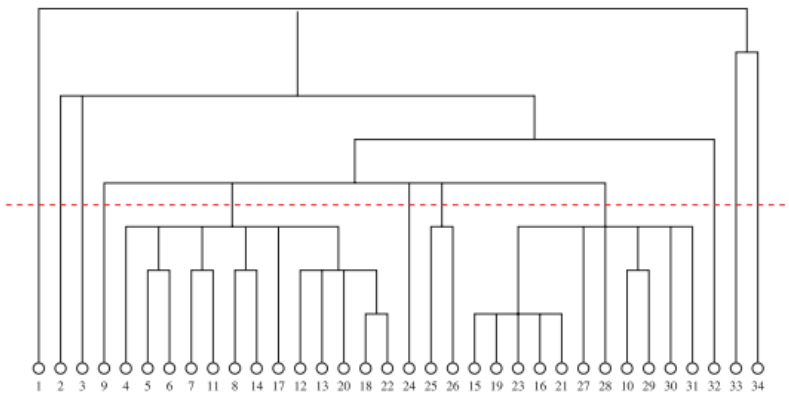
Girvan and Newman algorithm

Idea: Remove edges from the networks, with high "betweenness score", iteratively.

Motivation: Few edges between communities are bottlenecks. Traffic has to travel through them.

Algorithm

- Edge betweennes: # of geodesic paths between vertex pairs containing the edge.
- Remove edges with the highest betweennesses until no edges remains.
- Progress represented in dendogram:



Example: Girvan and Newman algorithm

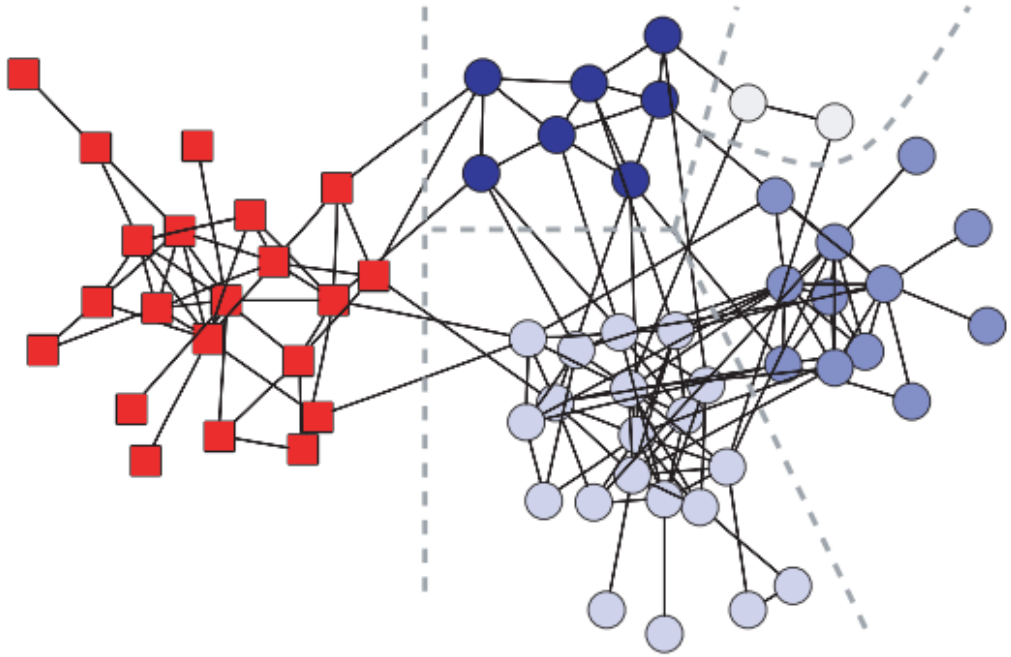


Fig. 4. Community structure in the social network of bottlenose dolphins assembled by Lusseau et al. [36,37], extracted using the algorithm of Girvan and Newman [1]. The squares and circles denote the primary split of the network into two groups and the circles are further subdivided into four smaller groups as shown. After Newman and Girvan [38].

Girvan and Newman algorithm II.

Problem: No guide how many communities to have.

Girvan and Newman algorithm II.

Problem: No guide how many communities to have.
Solution:

- Introduce again modularity:
  Q = fraction of edges within communities - expected value of the same quantity
- If $Q = 0$ community structure is not stronger than by random chance.
- Local peaks of Q during the algorithm indicates good divisions.

Runtime: Slow $O(m^2 n)$ or $O(n^3)$.

Girvan and Newman algorithm II.

Problem: No guide how many communities to have.
Solution:

- Introduce again modularity:
  Q = fraction of edges within communities - expected value of the same quantity
- If $Q = 0$ community structure is not stronger than by random chance.
- Local peaks of Q during the algorithm indicates good divisions.

Runtime: Slow $O(m^2 n)$ or $O(n^3)$.

Extensions:

- Monte Carlo estimate of betweennes Tyler at al.
- Local measure of betweennes (short loops) $O(m^4/n^2)$ Radachi et al.

Modularity: multiple communities

Shortcomings: two communities, using only leading eigenvector.

Modularity: multiple communities

Shortcomings: two communities, using only leading eigenvector.

Goal: Generalize to c communities.

$$S_{i,j} = \begin{cases} 1 & \text{if vertex } i \text{ belongs to community } j, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Then the modularity

$$Q = Tr(\mathbf{S}^T \mathbf{B} \mathbf{S}) = \sum_{j=1}^{n} \sum_{k=1}^{n} \beta_j (\mathbf{u}_j^T \mathbf{s}_k)^2,$$

where $\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ ($\mathbf{D}$ is diagonal with $D_{i,i} = \beta_i$)

Optimally: Choose mutually orthogonal $\mathbf{s}_1, ..., \mathbf{s}_{c-1}$ proportional to the leading eigenvectors with positive eigenvalues.

Modularity: multiple communities

Shortcomings: two communities, using only leading eigenvector.

Goal: Generalize to c communities.

$$S_{i,j} = \begin{cases} 1 & \text{if vertex } i \text{ belongs to community } j, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Then the modularity

$$Q = Tr(\mathbf{S}^T \mathbf{B} \mathbf{S}) = \sum_{j=1}^{n} \sum_{k=1}^{n} \beta_j (\mathbf{u}_j^T \mathbf{s}_k)^2,$$

where $\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ ($\mathbf{D}$ is diagonal with $D_{i,i} = \beta_i$)

Optimally: Choose mutually orthogonal $\mathbf{s}_1, ..., \mathbf{s}_{c-1}$ proportional to the leading eigenvectors with positive eigenvalues.

Problem: $s_i \in \{0, 1\}$ and may not be possible to find as many index vectors making positive contribution. Therefore it gives only an upper bound on the number of communities.

Modularity: multiple communities II

Generalization:

- Rewrite the modularity (for possible negative α):

$$Q = n\alpha + Tr[\mathbf{S}^T\mathbf{U}(\mathbf{D} - \alpha\mathbf{I})\mathbf{U}^T\mathbf{S}],$$

- Then define the $p \leq n$ dimensional vertex vector $\mathbf{r}_i$:

$$[\mathbf{r}_i]_j = \sqrt{\beta_j - \alpha}U_{i,j}$$

- Keeping the leading p eigenvalues

$$Q \approx n\alpha + \sum_{k=1}^{c}\sum_{j=1}^{p}\Big[\sum_{i\in G_k}[\mathbf{r}_i]_j\Big]^2 =: n\alpha + \sum_{k=1}^{c}|\mathbf{R}_k|^2.$$

Goal: Maximize the magnitude of vectors $\mathbf{R}_k = \sum_{i\in G_k}\mathbf{r}_i$ by dividing the vertices into groups.

Connections: It is also called the "Principal component analysis of networks".

Modularity: multiple communities II

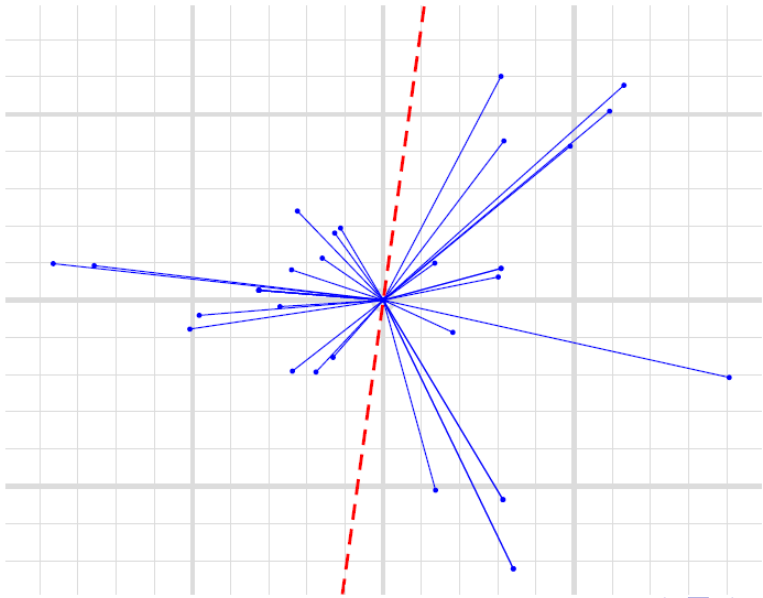
Maximizing the magnitude of $\mathbf{R}_k$:

- From the orthogonality of the eigenvectors we have

$$\sum_{k=1}^{c} \mathbf{R}_k = \sum_{i=1}^{n} \mathbf{r}_i = 0$$

- For $c = 2$: $\mathbf{R}_1$ $\mathbf{R}_2$ are equal magnitude and opposite directed.
- Removing vertex i from community k where $\mathbf{R}_k^T \mathbf{r}_i < 0$:

$$|\mathbf{R}_k - \mathbf{r}_i|^2 - |\mathbf{R}_k|^2 = |\mathbf{r}_i|^2 - 2\mathbf{R}_k^T \mathbf{r}_i > 0.$$



Conclusion

- Algorithms for bisection graphs with known community size (Laplacian spectral algorithm, The Kernighan-Lin algorithm)
- In lot of real-world example the community sizes are unknown (Modularity algorithm)
- Modularity matrix contains various information (anti-community structure, network correlation, community centrality)
- Furthermore in real world examples usually there are more communities (Girvan and Newman algorithm, generalized modularity algorithm)
- Modularity algorithm: connection with figuration model, PCA