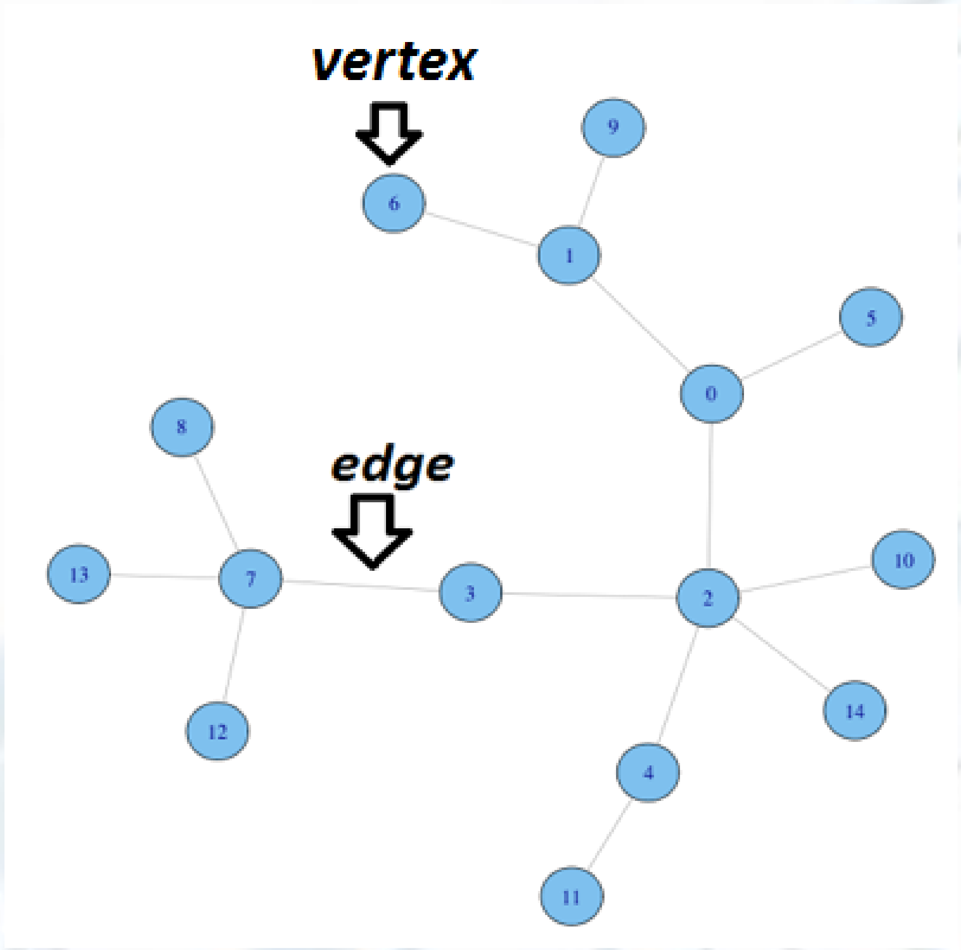
Networks properties study through a spectral method

A network is a set of points or ***vertices*** interlinked by ***edges***



A lot of natural fenomena can be described by networks, for example:

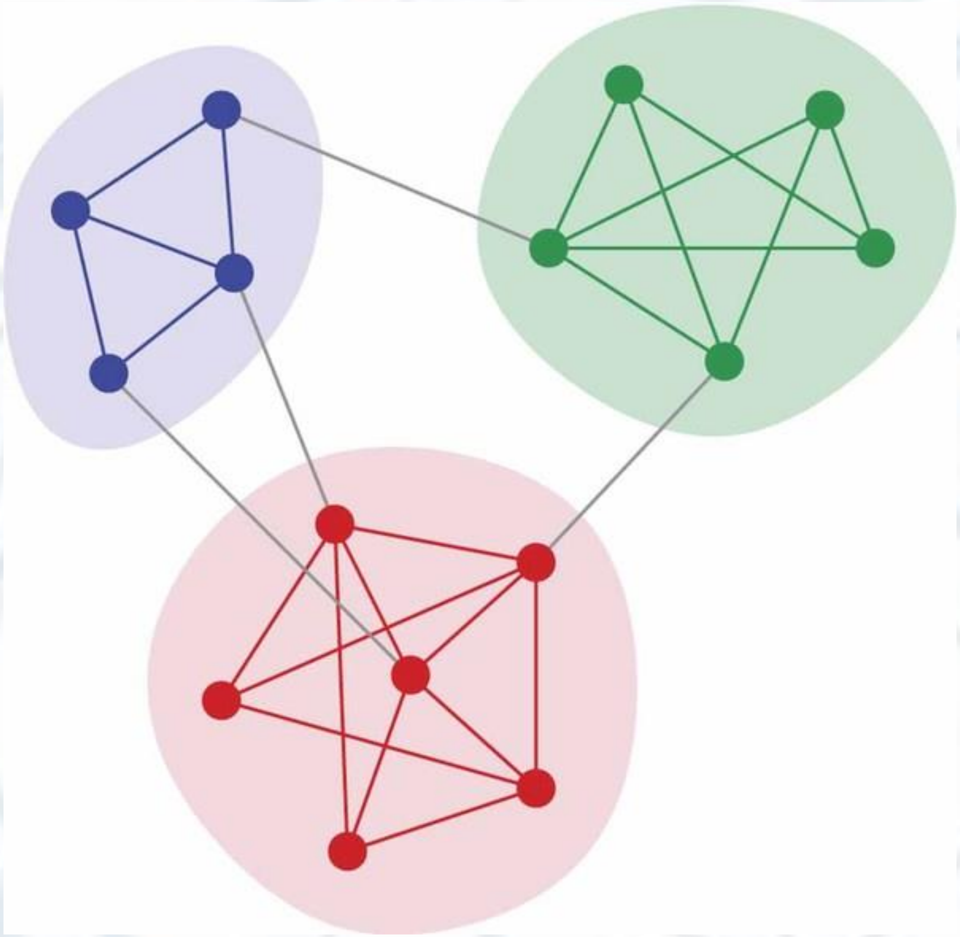
- In **neuroscience** the brain areas are represented by vertices and the anatomical connections by edges
- In **proteomics** the proteins are represented by vertices and the chemical interactions by edges
- In **financial markets** the investors, the issuers and the financial intermediaries are represented by vertices and the capital transfers by edges
- In **social networks** the individuals are represented by vertices while the social ties, the random knowledge, the labor relations and the family ties are represented by edges
- In **internet** the computers are represented by vertices and the connections between them by edges

Indeed the knowledge of the features of a network allows us to understand what will be the behaviour of the systems from it described

An important feature a network can own is a community structure

Community: vertices of a network partition so that the connections within the obtained groups are denser than outside them.

<u>More edges «inside» than «outside»</u>



But what is it the best partition?

M. E. J. Newman and M. Girvan, Finding and evaluating community structure in networks. Phys. Rev. E 69,026113 (2004).

Modularity: scalar objective function

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \boxed{\frac{d_i d_j}{2m}}\right]\delta_{g_i g_j}$$

Probability to find an edge between the vertices i and j in a random network

$$\begin{cases} 1 \text{ inside a community} \\ 0 \quad \text{otherwise} \end{cases}$$

Where

- d_i = vertex degree, i = # of edges linked to vertex i
- m = total number of edges within the network
- g_i community to which vertex i belongs
- A_{ij} elements of the Adjacency matrix $\mathbf{A}$ n x n

$$A_{ij} = \begin{cases} 1 \text{ if vertices } i \text{ and } j \text{ are linked by an edge} \\ 0 \text{ if vertices } i \text{ and } j \text{ are } \textbf{not} \text{ linked} \end{cases}$$

A modularity value is assigned to each partition of a network in communities. The partition with the highest Q value will be the best one.

M. E. J. Newman, Modularity and community structure in networks. Proc. Natl. Acad. Sci. USA 103, 8577–8582 (2006).

OBJECTIVE

We have to find the community structure which maximizes Q

But the complete numerical maximization of the modularity is an NP-hard problem, treatable only for networks with very reduced dimensions

Therefore we have to rely to an approximated heuristics optimization

There are many algorithms that can partition the vertices of a network

SUPERVISED

We fix the communities

UNSUPERVISED

The communities are fixed by the alghorithm

Here I will focus only on the supervised one

One of the most used is the k-*means algorithm*, as defined:

- We start by choosing k centroids in the space, initially
- We compute the distance of each point to each of the k centroids
- We associate a point to the k group according to what is its nearest k centroid
- We compute the k centroids of the new groups substituting the old centroids with the new ones, then we iterate the process
- Until the centroids stop to vary, the process continues

We want to improve the k-means method exploiting the matrix theory

- Lets define the modularity matrix $\mathbf{B}$ symmetric n x n so that

$$B_{ij} = A_{ij} - \frac{d_i d_j}{2m}$$

Substituting we get

$$Q = \frac{1}{2m} \sum_{ij} B_{ij} \, \delta_{g_i g_j}$$

Xiao Zhang and M. E. J. Newman, Multiway spectral community detection in networks.

- $\mathbf{B}$ is symmetric hence it can be always be written as an eigenvectors decomposition

$$B_{ij} = \sum_{l=1}^{n} \lambda_l U_{il} U_{jl}$$

Where
  - λ_l is an eigenvalue of $\mathbf{B}$
  - U_{il} is an element of the orthogonal matrix $\mathbf{U}$ whose columns are the respective eigenvectors
  - $n =$ number of vertices within the network
- If $k =$ number of communities within the network

$$\delta_{g_i g_j} = \sum_{s=1}^{k} \delta_{s g_i} \delta_{s g_j}$$

- Lets define a set of n vertex vectors $\mathbf{r}_\mathrm{i}$ of components $[\mathbf{r}_\mathrm{i}]_l = \sqrt{\lambda_l} U_{il}$ which is equivalent to assign to each vertex an $\mathbf{r}_\mathrm{i}$ vector

- Lets define the group vectors $\mathbf{R}_\mathrm{s} = \sum_{i\in s} \mathbf{r}_i$, one for each community s
- Hence we obtain, after a few simple calculations

$$\boxed{Q = \frac{1}{2m}\left|\sum_s \mathbf{R}_\mathrm{s}\right|^2}$$

- Imagine moving a vertex i from a community s to another t one
  - before moving the group vectors of the communities are respectively $\mathbf{R}_\mathrm{s} + \mathbf{r}_i$ and $\mathbf{R}_\mathrm{t}$
  - after moving they are $\mathbf{R}_s$ e $\mathbf{R}_\mathrm{t} + \mathbf{r}_i$

$$\Delta Q = \frac{1}{2m}[|\mathbf{R}_\mathrm{s}|^2 + |\mathbf{R}_\mathrm{t} + \mathbf{r}_i|^2 - |\mathbf{R}_s + \mathbf{r}_i|^2 - |\mathbf{R}_t|^2] =$$

$$= \frac{1}{m}[\mathbf{R}_t^T\mathbf{r}_i - \mathbf{R}_s^T\mathbf{r}_i]$$

- Since we want to maximize the modularity there must be $\boxed{\Delta Q > 0}$

$$\Downarrow$$

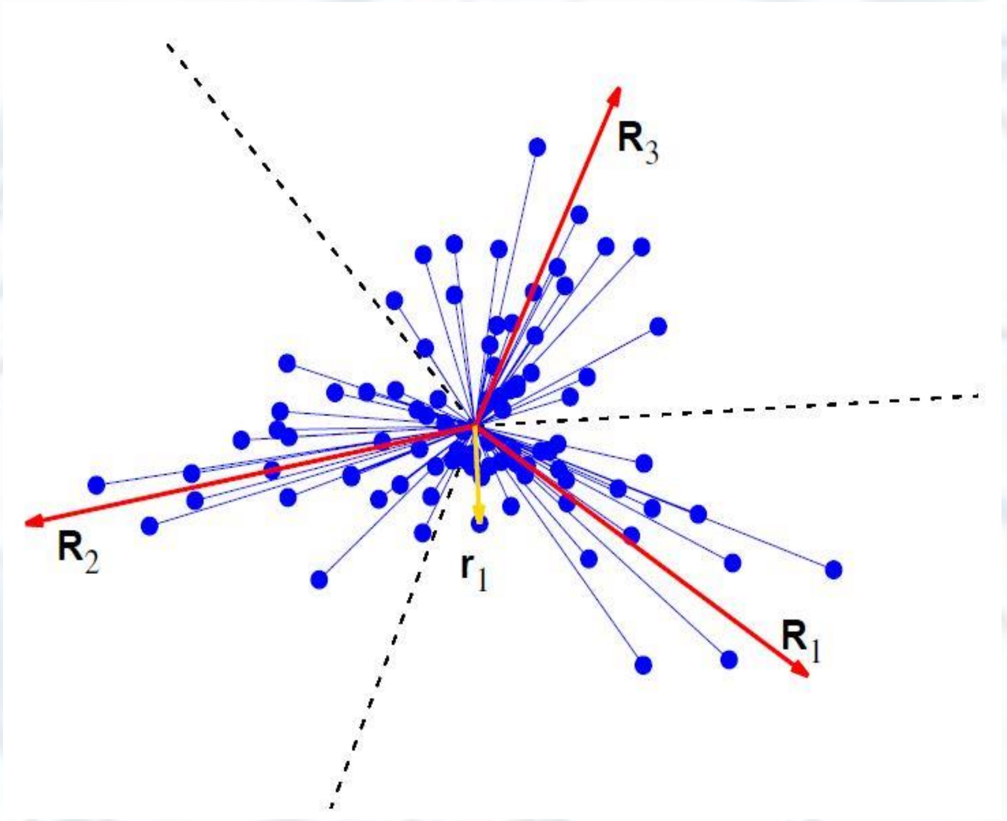
It's equivalent to assign the i vertex to the community whose group vector has the higher dot product with $\mathbf{r}_i$

Therefore the dot product between $\mathbf{r}_i$ and a group vector defines the analogue of the point-centroid distance

Now we have all the elements to define a new partitioning algorithm

VECTOR PARTITIONING ALGORITHM

1. Choose an initial set of group vectors $\mathbf{R}_s$, one for each of the k communities.
2. Compute the dot product $\mathbf{R}_s^T \mathbf{r}_i$ for all the i vertices and for all the s groups
3. Assign each vertex to the community with which it has the major dot product
4. Update the group vectors using the definition $\mathbf{R}_s = \sum_{i \in s} \mathbf{r}_i$
5. Repeat from step 2 until the group vectors stop changing



TESTS

- Synthetic networks

  Lets define

  – *degree-corrected stochastic block model (D-C SBM)*

    - Model that generates networks with a community structure.
    - The edges are placed between i, j pairs of vertices with probability $d_i d_j \omega_{st}$

Lets define ω_{st} as $\omega_{st} = (1-\delta)\omega_{st}^{random} + \delta\omega_{st}^{planted}$

So that $\omega_{st}^{random} = \frac{1}{2m}$, $\omega_{st}^{planted} = \frac{\delta_{st}}{\sum_{i\in s} d_i}$ and δ parameter $\in [0,1]$

In this way the probability of an edge between the two vertices i, j is

$$\begin{cases} \frac{d_i d_j}{2m} & \text{per } \delta = 0 \\ \frac{d_i d_j}{\sum_{i\in s} d_i} & \text{per } \delta = 1 \text{ e } s = t \\ 0 & \text{per } \delta = 1 \text{ e } s \neq t \end{cases}$$

It corresponds to a ***random*** distribution of the edges with no communities within the network

- *Mutual information*
  - We have two sets X, Y of data
  - X: variable which contains the assingnments of the vertices to the communities in the stochastic block model
  - Y: variable which contains the assingnments of the vertices to the communities the partitioning algorithm has found

  Hence

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}$$

  Where
  - $p(x,y)$ is the frequency of x and y within the dataset
  - $p(x)$ and $p(y)$ marginal probabilities of x and y
- *Normalized mutual information or NMI*

$$NMI(X;Y) = \frac{I(X;Y)}{\frac{1}{2}[H(X) + H(Y)]}$$

  Where
  - $H(X)$ and $H(Y)$ are the hentropies of the sets X and Y

$$NMI = \begin{cases} 0 \text{ if } X, Y \quad\quad \text{are not correlated} \\ 1 \text{ if } X, Y \text{ are perfectly correlated} \end{cases}$$

Example
Synthetic networks generated with a D-C SBM

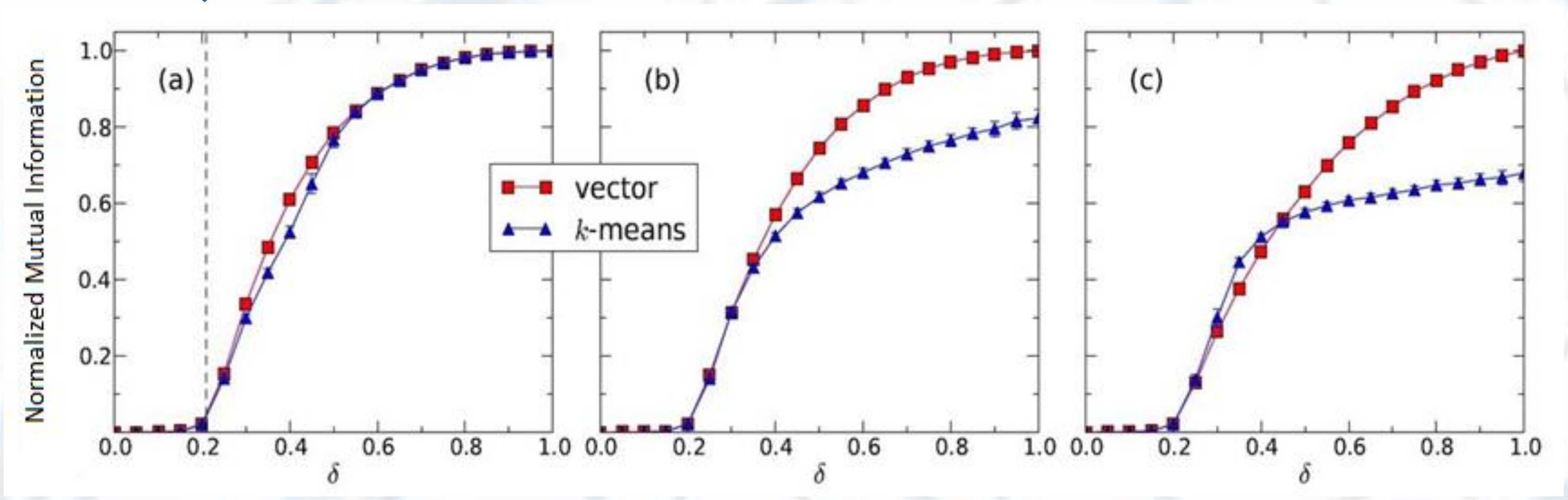
- $n = 3600$ vertices
- Subdivided into $k = 3$ communities
- The 50% of the vertices belonging to each community has degree equal to 10
- The 50% of the vertices belonging to each community has degree equal to 30
- Fig. (a): equal dimension communities, each one composed of 1200 vertices
- Fig. (b): communities containing respectively 1800, 1200 and 600 vertices
- Fig. (c): communities containing respectively 2400, 900 e 300 vertices
- Each point is weighted average of 100 networks

Lets test the partitioning algorithm and the k-*means* algorithm on this networks

- Vector partitioning
  - $\delta \to 1$ $NMI \to 1$ in (a), (b), (c)
  - $\delta \to 0$ $NMI \to 0$ in (a), (b), (c)
- k-means
  - $\delta \to 1$ $NMI \to 1$ in (a) mentre $NMI < 1$ in (b), (c)
  - $\delta \to 0$ $NMI \to 0$ in (a), (b), (c)

The vector patitioning algorithm obtains better results then the k-means algorithm in almost all cases

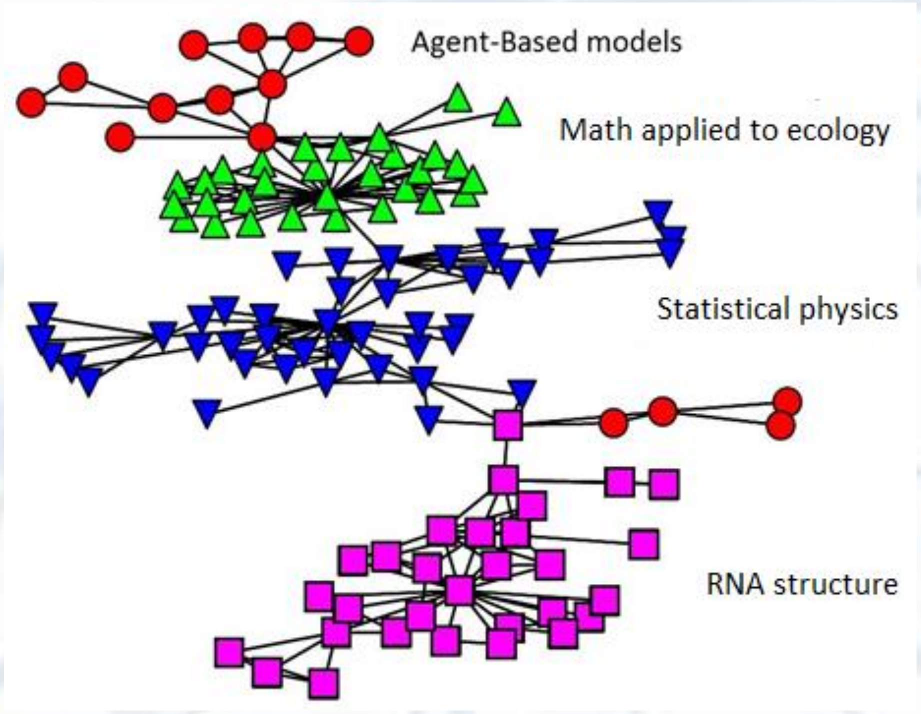
Detection threshold under which both methods fail



- Real world networks

Example

- The vertices of the network represent the scientists working at the Santa Fe Institute
- The edges indicates two scientists who were co-authors of an article
- There are four main research fields
- We analyze the network through the partitioning algorithm
- Differenti colors and shapes indicate the communities discovered through the vector partitioning algorithm



The results show that the four research areas roughly correspond to the communities the algorithm has found

Four vertices were misclassified. This highlights a weakness of the algorithm: the main feature that characterizes this vertices is the very small module of their respective vertex vectors, hence they do not belong in a clear way to any community.

It is possible to solve this problem combining the algorithm with a “fine tuning” process

CONCLUSIONS

To define the number of communities within a network taking into account only the features of edges and vertices is an interesting problem still open

- *Observations*
  - In the synthetic networks case we have proved that the vector partitioning algorithm always obtains better performance than the concurrent spectral algorithm, which makes use of the k-means clustering
  - We have provided an application example to the real world networks due to prove its effectiveness
  - The algorithm requires the prior knowledge of the number of communities within the network
  - Although the algorithm is simple and efficient it is only heuristic
- *Future goals*
  - To develop a method which combines the vector partitioning algorithm with other elementary methods in order to get exact results

Bibliography

[1] Xiao Zhang and M. E. J. Newman, Multiway spectral community detection in networks.
[2] M. E. J. Newman and M. Girvan, Finding and evaluating community structure in networks. Phys. Rev. E 69, 026113 (2004).
[3] M. E. J. Newman, Fast algorithm for detecting community structure in networks. Phys. Rev. E 69, 066133 (2004).
[4] A. Clauset, M. E. J. Newman, and C. Moore, Finding community structure in very large networks. Phys. Rev. E 70, 066111 (2004).
[5] R. Guimer`a, M. Sales-Pardo, and L. A. N. Amaral, Modularity from fluctuations in random graphs and complex networks. Phys. Rev. E 70, 025101 (2004).
[6] J. Duch and A. Arenas, Community detection in complex networks using extremal optimization. Phys. Rev. E 72, 027104 (2005).
[7] L. Shuzhuo, C. Yinghui, D. Haifeng, and M. W. Feldman, A genetic algorithm with local search strategy for improved detection of community structure. Complexity 15(4), 53–60 (2010).
[8] M. E. J. Newman, Modularity and community structure in networks. Proc. Natl. Acad. Sci. USA 103, 8577–8582 (2006).
[9] R. R. Nadakuditi and M. E. J. Newman, Graph spectra and the detectability of community structure in networks. Phys. Rev. Lett. 108, 188701 (2012).
[10] M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74, 036104 (2006).
[11] T. Richardson, P. J. Mucha, and M. A. Porter, Spectral tripartitioning of networks. Phys. Rev. E 80, 036111 (2009).
[12] U. Elsner, Graph partitioning—a survey. Technical Report 97-27, Technische Universit¨at Chemnitz (1997).
[13] M. E. J. Newman, Spectral methods for network community detection and graph partitioning. Phys. Rev. E 88, 042822 (2013).
[14] S. Onn and L. J. Schulman, The vector partition problem for convex objective functions. Mathematics of Operations Research 26, 583–590 (2001).
[15] G.Wang, Y. Shen, and M. Ouyang, A vector partitioning approach to detecting community structure in complex networks. Computers and Mathematics with Applications 55, 2746–2752 (2008).
[16] B. Karrer and M. E. J. Newman, Stochastic blockmodels and community structure in networks. Phys. Rev. E 83, 016107 (2011).
[17] M. Girvan and M. E. J. Newman, Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA 99, 7821–7826 (2002).
[18] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, Random graphs with arbitrary degree distributions and their applications. Phys. Rev. E 64, 026118 (2001).

APPENDIX

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{d_i d_j}{2m}\right]\delta_{g_i g_j}$$

$$B_{ij} = A_{ij} - \frac{d_i d_j}{2m}$$

Substituting we get $Q = \frac{1}{2m}\sum_{ij} B_{ij}\,\delta_{g_i g_j}$

By definition $\sum_i B_{ij} = \sum_i A_{ij} - \sum_i \frac{d_i d_j}{2m}$

$\sum_i A_{ij} = d_j$ e $\sum_i d_i = 2m$

so $\sum_i B_{ij} = d_j - d_j \cdot 1 = 0$

Therefore by the eigenvalue definition **1** = (1,1,1,....) is an eigenvector of **B** belonging to the zero eigenvalue

B is symmetric hence it can always be written as an eigenvectors decomposition

$$B_{ij} = \sum_{l=1}^{n} \lambda_l U_{il} U_{jl}$$

$$\delta_{g_i g_j} = \sum_{s=1}^{k} \delta_{s g_i} \delta_{s g_j}$$

we can rewrite the modularity as

$Q = \frac{1}{2m} \sum_{l=1}^{n} \lambda_l \sum_{s=1}^{k} \left[\sum_i U_{il} \delta_{s g_i} \right]^2$ considering only the positive eigenvalues

$Q = \frac{1}{2m} \sum_{l=1}^{p} \lambda_l \sum_{s=1}^{k} \left[\sum_i U_{il} \delta_{s g_i} \right]^2$ for $p < n$

$$Q = \frac{1}{2m} \sum_{s=1}^{k} \sum_{l=1}^{p} \left[\sum_i \sqrt{\lambda_l} U_{il}\, \delta_{s g_i} \right]^2$$

$[\mathbf{r}_\mathrm{i}]_l = \sqrt{\lambda_l} U_{il}$ hence

$$Q = \frac{1}{2m} \sum_{s=1}^{k} \left| \sum_{i \in s} \mathbf{r}_\mathrm{i} \right|^2$$

$\mathbf{R}_\mathrm{s} = \sum_{i \in s} \mathbf{r}_i$ so

$$Q = \frac{1}{2m} \left| \sum_{s} \mathbf{R}_\mathrm{s} \right|^2$$

- In reality it's necessary to choose only $k-1$ group vectors in the partitioning algorithm, indeed

  $\mathbf{1}$ is an eigenvector of $\mathbf{B}$, the eigenvectors are l.i hence

  $\mathbf{1} \cdot \boldsymbol{x} = x_1 + x_2 + \cdots = 0$ with $\boldsymbol{x} = (x_1, x_2, \dots)$ eigenvector of $\mathbf{B}$, so

  $\sum_{i=1}^{n} [\mathbf{r}_\mathrm{i}]_l = \sqrt{\lambda_l} \sum_{i=1}^{n} U_{il} = 0$ thus

  $\sum_{i=1}^{n} \mathbf{r}_\mathrm{i} = 0$ e $\sum_s \mathbf{R}_\mathrm{s} = \sum_s \sum_{i \in s} \mathbf{r}_\mathrm{i} = \sum_{i=1}^{n} \mathbf{r}_\mathrm{i} = 0$ and therefore

$$\boxed{\mathbf{R}_\mathrm{k} = -\mathbf{R}_1 - \mathbf{R}_2 - \cdots - \mathbf{R}_{\mathrm{k}-1}}$$