Community structure in networks

Argimiro Arratia & Ramon Ferrer-i-Cancho

Universitat Politècnica de Catalunya

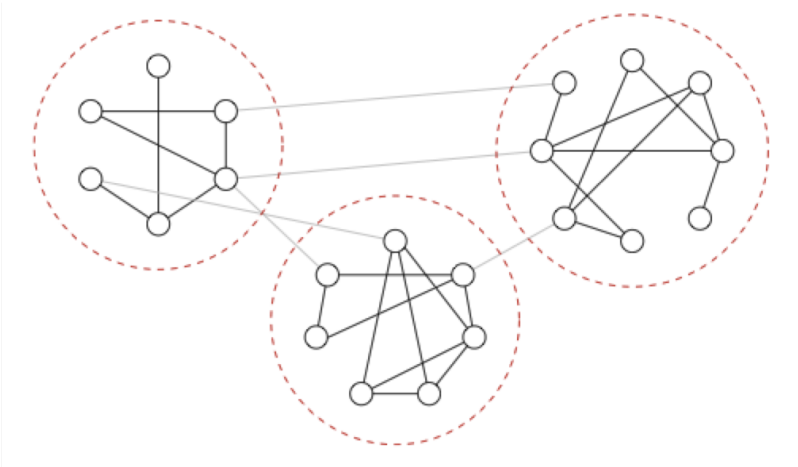
Version 0.6
Complex and Social Networks (2020-2021)
Master in Innovation and Research in Informatics (MIRI)

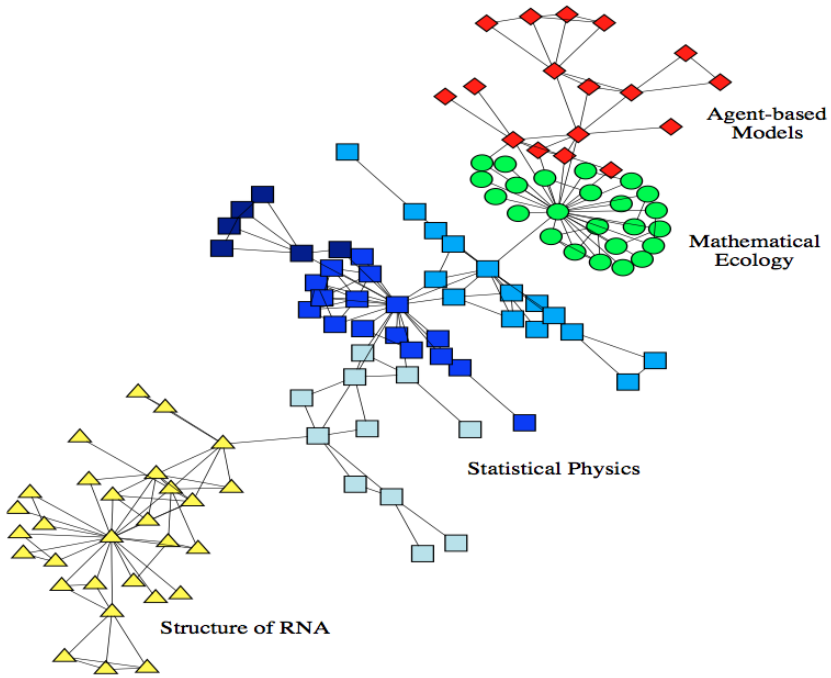
Instructors

- Ramon Ferrer-i-Cancho, rferrericancho@cs.upc.edu, `http://www.cs.upc.edu/~rferrericancho/`
- Argimiro Arratia, argimiro@cs.upc.edu, `http://www.cs.upc.edu/~argimiro/`

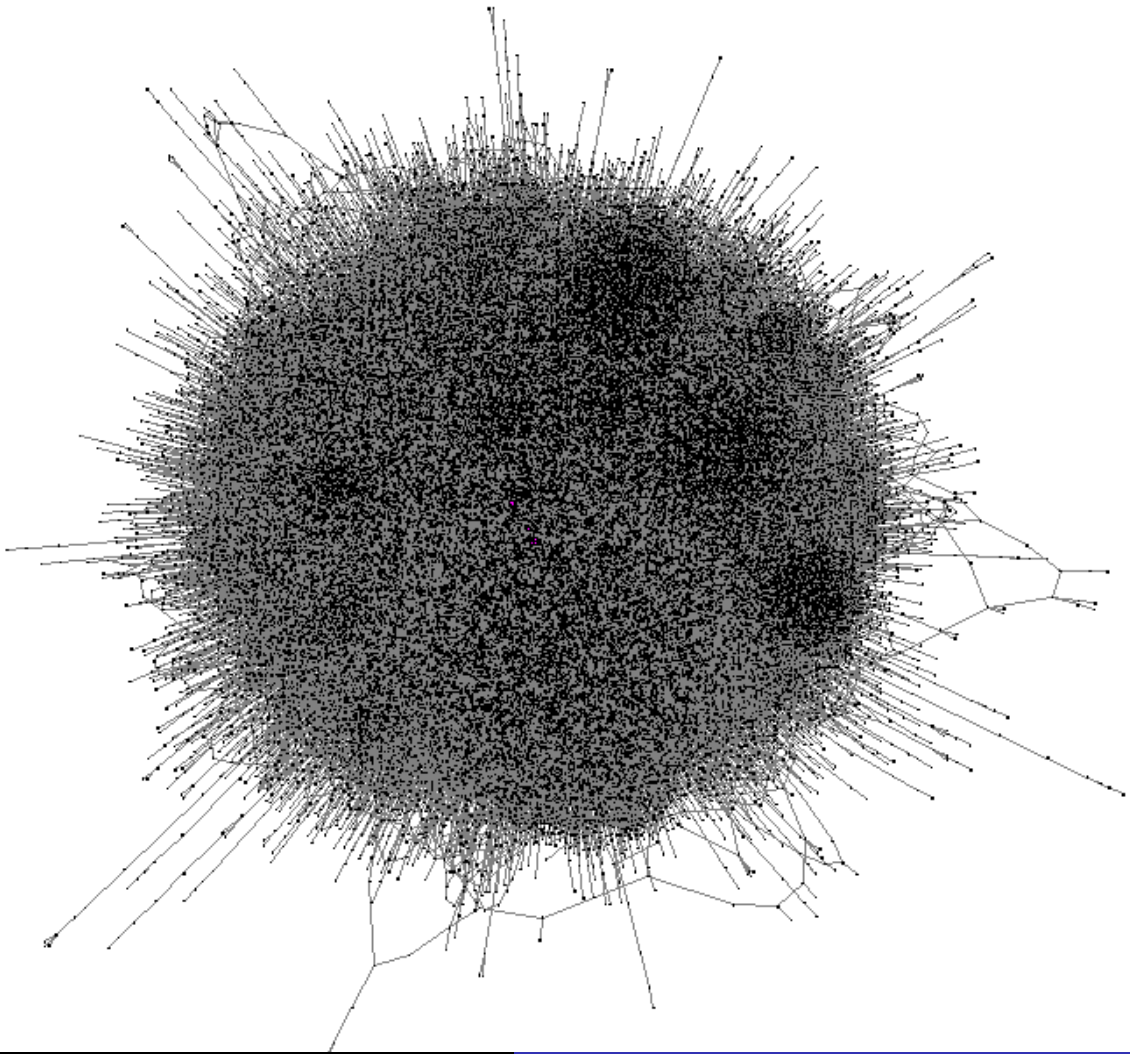
Please go to `http://www.cs.upc.edu/~csn` for all course's material, schedule, lab work, etc.

What is community structure?



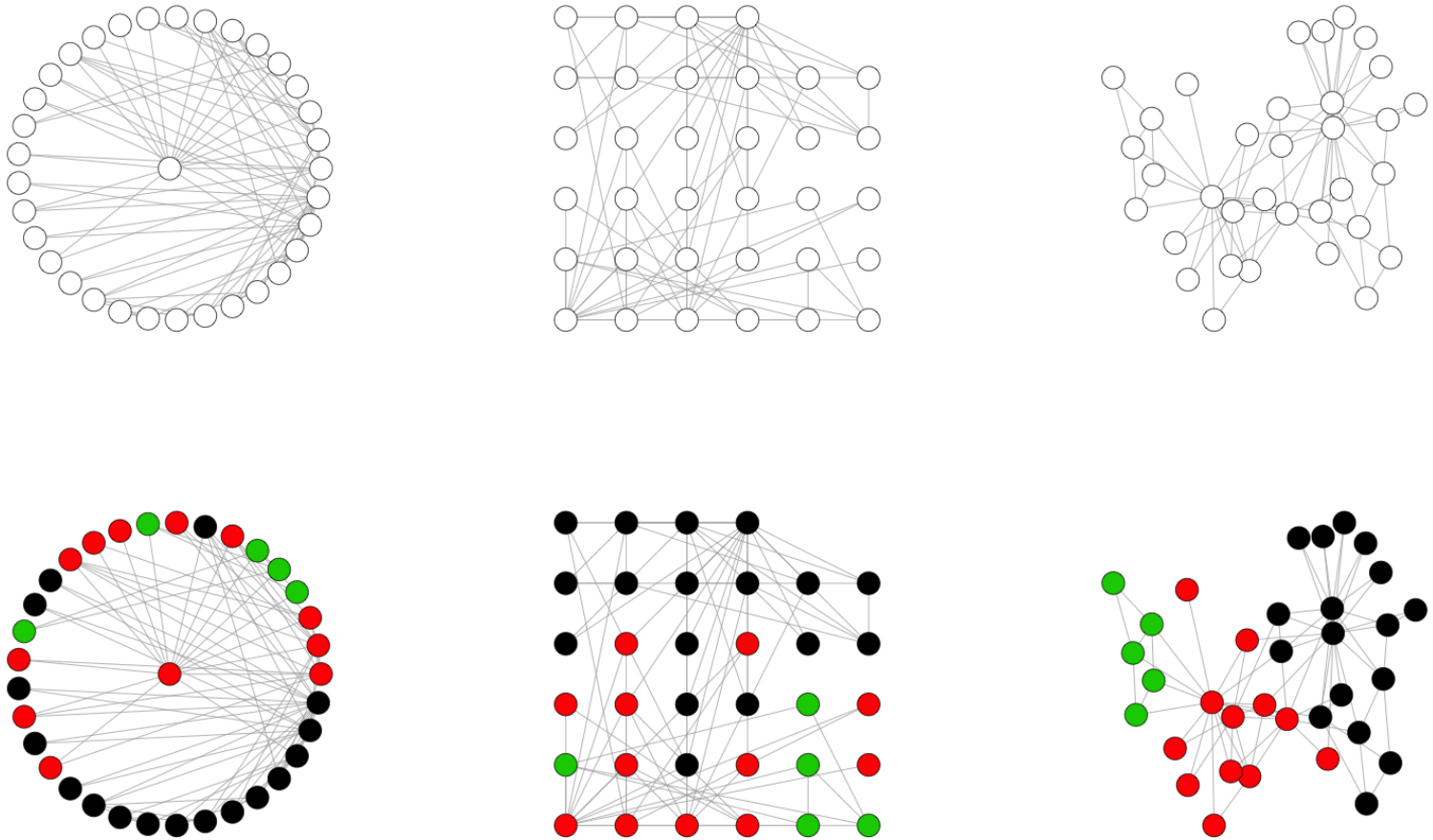


Why is community structure important?



.. but don't trust visual perception

it is best to use objective algorithms



Contents

Clustering algorithms (General outlook)

Clustering algorithms are either:

Hierarchical

- Agglomerative: begin with singleton groups and join successively by similarity. E.g. Lovain algorithm
- Divisive: begin with one group containing all points and divide successively. E.g. Girvan-Newman

Partitional separate points in arbitrary number of groups and exchange elements according to similarity. E.g k-means, graph partition.

Clustering algorithms (General outlook)

Similarity

It is desirable that it has the properties of a distance metric (except possibly for triangle inequality which may not hold if graph is not complete).

- $d(x, y) \geq 0$ and $d(x, d) = 0$
- $d(x, y) = d(y, x)$
- $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality)

This is to guarantee convergence of clustering algorithms, usually based on greedy selection. If a distance $d(x, y)$ is considered then we talk about *dissimilarity*: high values $d(x, y)$ mean low similarity.

Clustering algorithms (General outlook)

If want to interpret high value of similarity as high similarity, and we are working with distance metric $d(x, y)$, the consider its inverse: $s(x, y) = 1/d(x, y)$ or $1/d(x, y) + 0.5$.

NB: We are here concern with clustering elements with an already defined rule of association (i.e. networks); hence similarity will reflect some structural property of the network. Other form of clustering (in statistical analysis) is on elements described by features from which one defines a *similarity network* (complete graph).

Similarity measures w_{ij} for nodes I

When network cannot be embedded in Euclidean space and similarity must be inferred from the adjacency relation between vertices (implicit similarity)

Let **A** be the adjacency matrix of the network, i.e. $A_{ij} = 1$ if $(i, j) \in E$ and 0 otherwise.

- **Jaccard index:**

$$w_{ij} = \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|} = \frac{\sum_k A_{ik} A_{kj}}{\sum_k (A_{ik} + A_{jk})}$$

where $\Gamma(i)$ is the set of neighbors of node i

Similarity measures w_{ij} for nodes II

- **Cosine similarity:** (From the equation $\mathbf{xy} = |\mathbf{x}||\mathbf{y}|\cos\theta$)

$$w_{ij} = \frac{\sum_k A_{ik}A_{kj}}{\sqrt{\sum_k A_{ik}^2}\sqrt{\sum_k A_{jk}^2}} = \frac{n_{ij}}{\sqrt{k_i k_j}} \quad \text{(recall } A_{ij} = 1 \text{ or } 0\text{)}$$

  where:
  - $n_{ij} = |\Gamma(i) \cap \Gamma(j)| = \sum_k A_{ik}A_{kj}$, and
  - $k_i = \sum_k A_{ik}$ is the degree of node i
- **Another normalization for n_{ij}:** the idea is to normalize by the *expected* number of common neighbors, if neighbors were chosen uniformly at random. This is approximately $k_i k_j / n$. And so

$$w_{ij} = \frac{n_{ij}}{k_i k_j / n} = n\frac{\sum_k A_{ik}A_{kj}}{\sum_k A_{ik}\sum_k A_{jk}}$$

Similarity measures w_{ij} for nodes III

- **Euclidean distance:** or rather Hamming distance since A is binary (a dissimilarity)

$$d_{ij} = \sum_k (A_{ik} - A_{jk})^2$$

- **Normalized Euclidean distance:**[1] (also a dissimilarity)

$$d_{ij} = \frac{\sum_k (A_{ik} - A_{jk})^2}{k_i + k_j} = 1 - 2\frac{n_{ij}}{k_i + k_j}$$

- **Pearson correlation coefficient**

$$r_{ij} = \frac{cov(A_i, A_j)}{\sigma_i \sigma_j} = \frac{\sum_k (A_{ik} - \mu_i)(A_{jk} - \mu_j)}{n\sigma_i \sigma_j}$$

where $\mu_i = \frac{1}{n}\sum_k A_{ik}$ and $\sigma_i = \sqrt{\frac{1}{n}\sum_k (A_{ik} - \mu_i)^2}$

[1] Uses the idea that maximum value of d_{ij} is when there are no common neighbors and then $d_{ij} = 1$

Similarity measures for sets of nodes

- Single linkage: $s_{XY} = \min_{x \in X, y \in Y} s_{xy}$
- Complete linkage: $s_{XY} = \max_{x \in X, y \in Y} s_{xy}$
- Average linkage: $s_{XY} = \frac{\sum_{x \in X, y \in Y} s_{xy}}{|X| \times |Y|}$
- Ward (or minimum variance): $s_{XY} = \frac{|X| \times |Y|}{|X| + |Y|} ||c_x - c_y||^2$,
  where c_x is the centroid of X:
  $\forall u, v \in X,\ ||u - c_x||^2 \leq ||u - v||^2$

Notes on similarity measures for sets of nodes

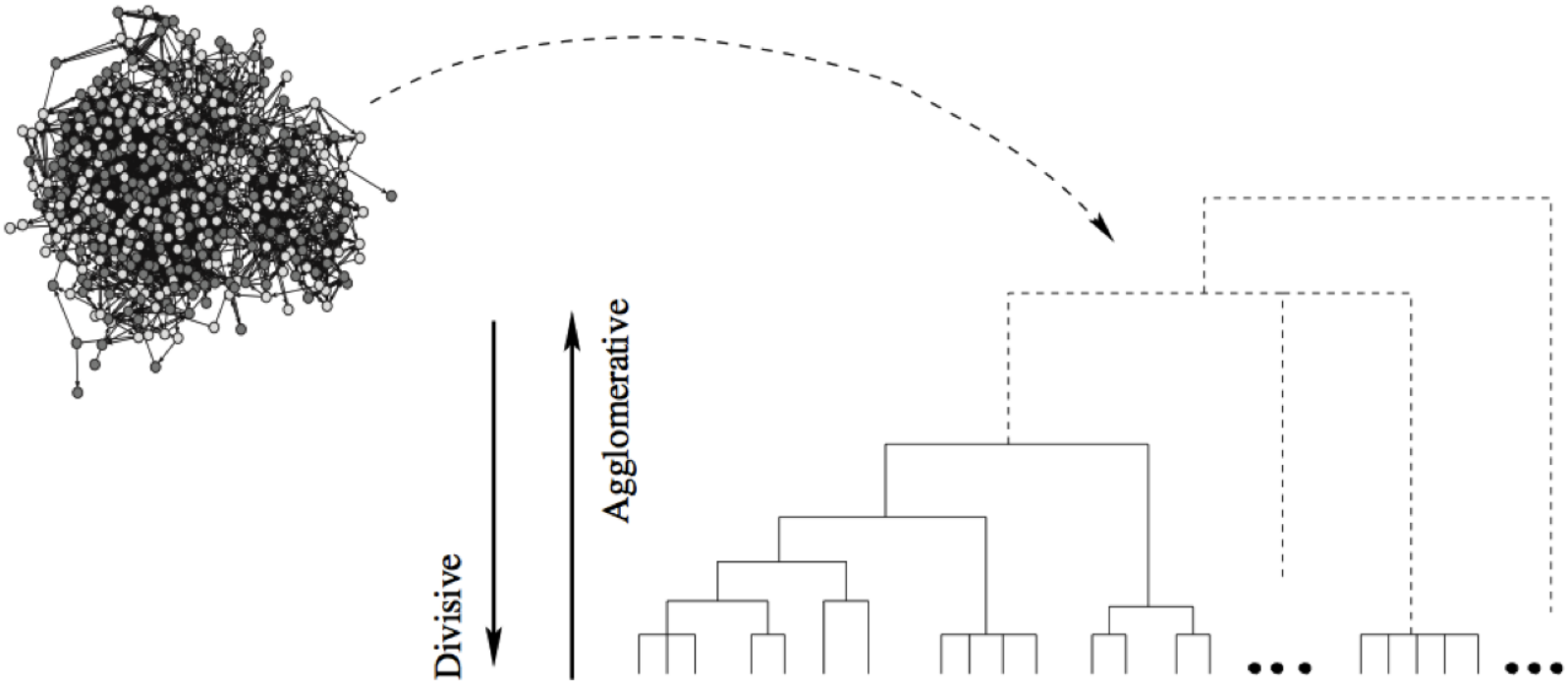
Ward's method says: "the distance between two clusters X and Y is how much the sum of squares will increase when we merge them".
In math:

$$\Delta(X,Y) = \sum_{i \in X \cup Y} ||x_i - c_{X \cup Y}||^2 - \sum_{i \in X} ||x_i - c_X||^2 - \sum_{i \in Y} ||x_i - c_Y||^2$$

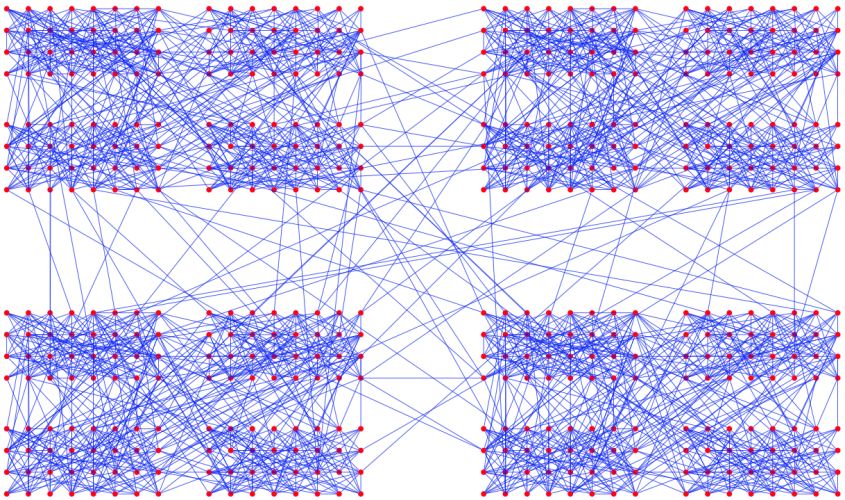
- single linkage : tends to make too small (in size) clusters
- complete: too big and fewer clusters
- average : more or less regular
- Ward's : tends to minimise the total within cluster variance

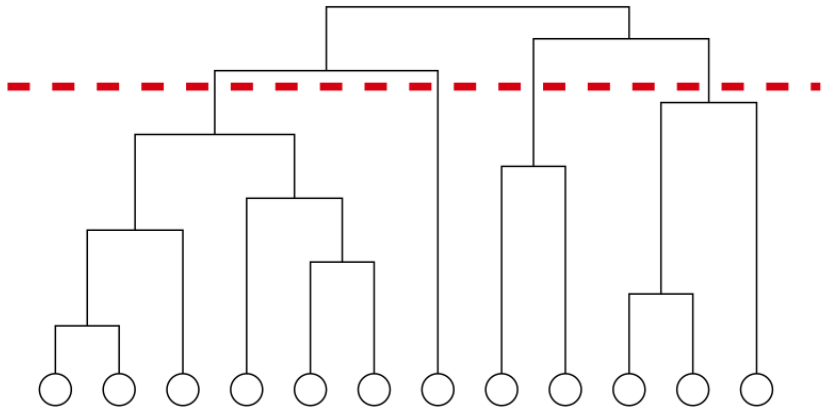
Hierarchical clustering

From hairball to *dendogram*



Suitable if input network has hierarchical structure





Agglomerative hierarchical clustering [Newman, 2010]

Ingredients

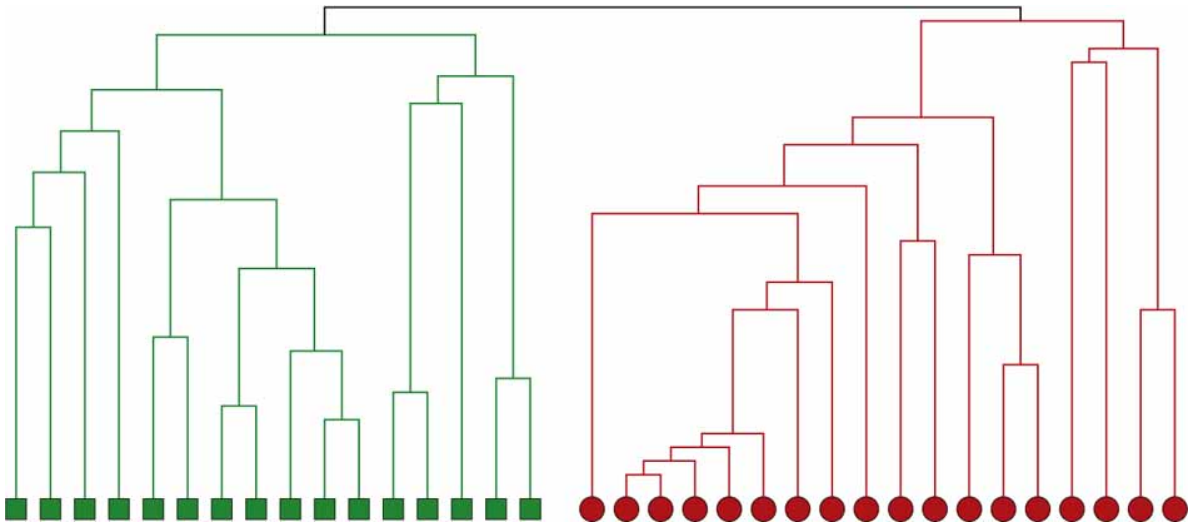
- Similarity measure between nodes
- Similarity measure between *sets of nodes*

Pseudocode

1. Assign each node to its own cluster
2. Find the cluster pair with highest similarity and join them together into a cluster
3. Compute new similarities between new joined cluster and others
4. Go to step 2 until all nodes form a single cluster
5. Select clustering (cut the tree at desired level)

Agglomerative hierarchical clustering on Zachary's network

Using average linkage



AHC on IBEX's stock daily returns (1/12/2008–1/2/2009)

Explicit similarity graph [Arratia, 2014]

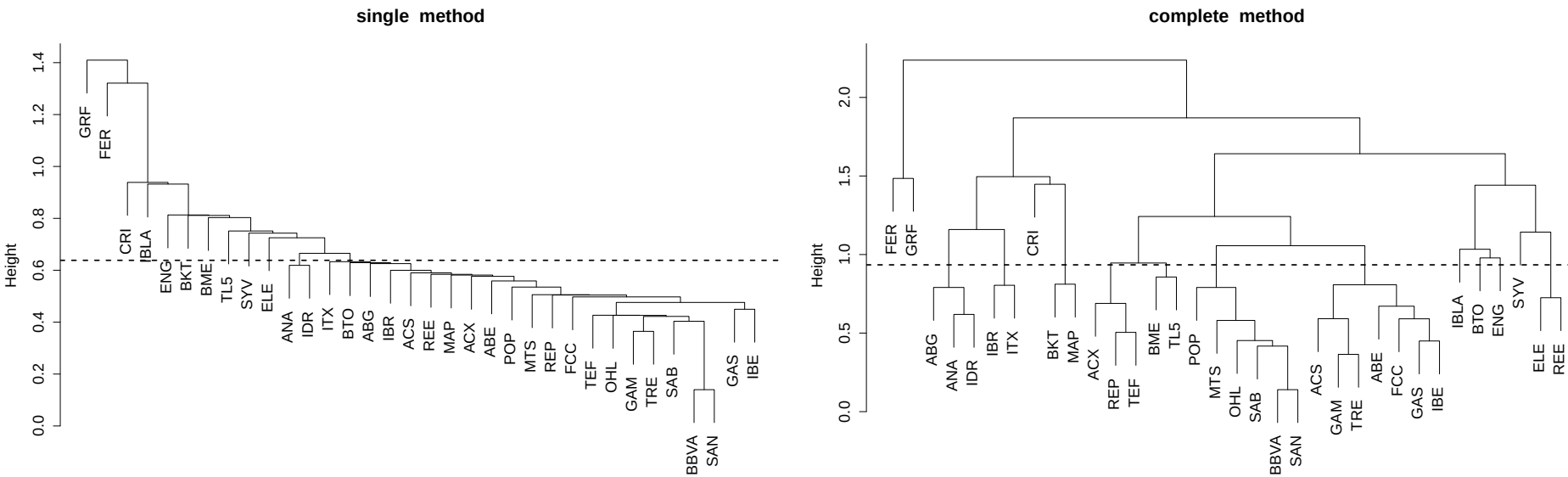


Figure: Dendrograms for single and complete inter-cluster linkages and dissimilarity measure $2(1 - \rho(\boldsymbol{x}, \boldsymbol{y}))$.

AHC on IBEX's stock daily returns (1/12/2008–1/2/2009)

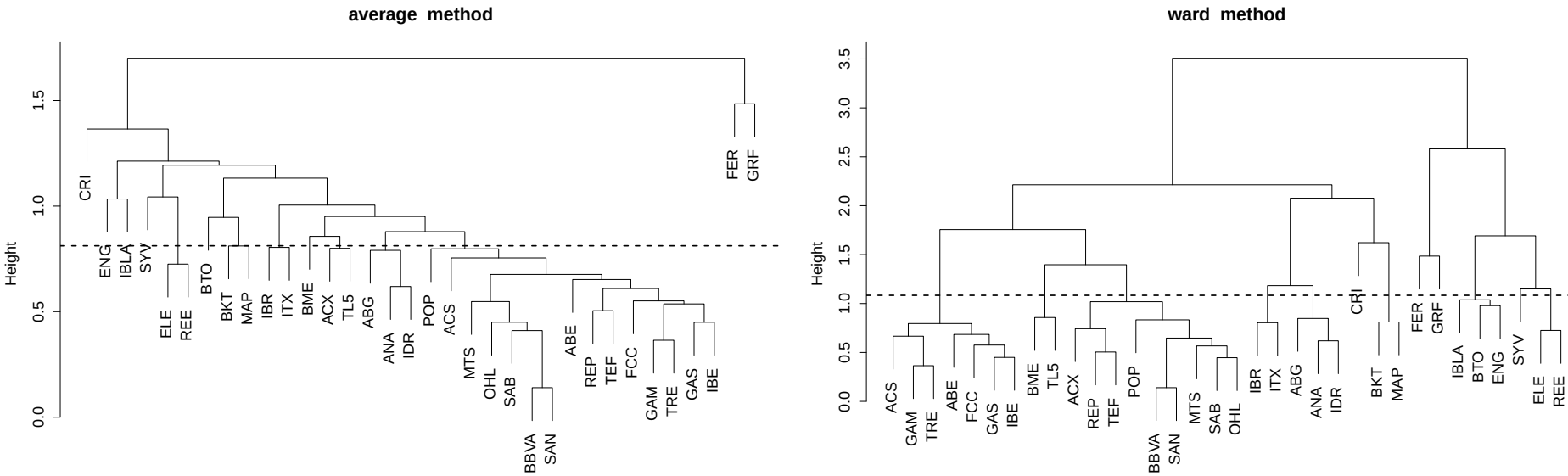


Figure: Dendrograms for average and Ward inter-cluster linkages and dissimilarity $2(1 - \rho(\boldsymbol{x}, \boldsymbol{y}))$.

Contents

Main idea

A community is *dense* in the inside but *sparse* w.r.t. the outside

No universal definition! But some ideas are:

- A community should be *densely connected*
- A community should be *well-separated* from the rest of the network
- Members of a community should be *more similar* among themselves than with the rest

Most common..

nr. of intra-cluster edges $>$ nr. of inter-cluster edges

Some definitions

Let $G = (V, E)$ be a network with $|V| = n$ nodes and $|E| = m$ edges. Let C be a subset of nodes in the network (a "cluster" or "community") of size $|C| = n_c$. Then

- *intra-cluster density*:

$$\delta_{int}(C) = \frac{\text{nr. internal edges of } C}{n_c(n_c - 1)/2}$$

- *inter-cluster density*:

$$\delta_{ext}(C) = \frac{\text{nr. inter-cluster edges of } C}{n_c(n - n_c)}$$

A community should have $\delta_{int}(C) > \delta(G)$, where $\delta(G)$ is the average edge density of the whole graph G, i.e.

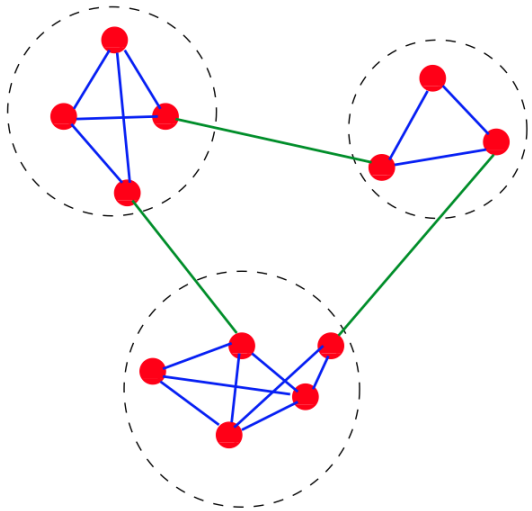
$$\delta(G) = \frac{\text{nr. edges in } G}{n(n-1)/2}$$

Most algorithms search for tradeoffs between *large* $\delta_{int}(C)$ and *small* $\delta_{ext}(C)$

- e.g. optimizing $\sum_C \delta_{int}(C) - \delta_{ext}(C)$ over all communities C

Define further:

- $m_c =$ nr. edges within cluster $C = |\{(u,v)|u,v \in C\}|$
- $f_c =$ nr. edges in the frontier of $C = |\{(u,v)|u \in C, v \notin C\}|$



- $n_{c_1} = 4, m_{c_1} = 5, f_{c_1} = 2$
- $n_{c_2} = 3, m_{c_2} = 3, f_{c_2} = 2$
- $n_{c_3} = 5, m_{c_3} = 8, f_{c_3} = 2$

Quality criteria I

Community scoring functions (i.e. characterize how community-like is the connectivity structure of set of nodes) can be group in four classes (measures in same class are highly correlated [Yang and Leskovec, 2012]):

Quality criteria II

(A) Based on internal connectivity (high is best)

- **Triangle participation ratio**: fraction of nodes in C that belong to a triad,

$$\frac{|\{u : u \in C \text{ and } \{(w,v) \in E : w,v \in C, (u,w),(u,v) \in E\} \neq \emptyset\}|}{n_c}$$

- **Internal density**: a.k.a. "intra-cluster density", or fraction of edges inside the cluster, $\frac{m_c}{n_c(n_c-1)/2}$
- Other: **edges inside**, **average degree**, **fraction over median degree**.

Quality criteria III

(B) Based on external connectivity (low is best)

- **expansion**: nr of edges per node leaving the cluster $\frac{f_c}{n_c}$
- **cut ratio**: a.k.a. "inter-cluster density": fraction of existing edges leaving the cluster, $\frac{f_c}{n_c(n-n_c)}$

Quality criteria IV

(C) Combine internal and external connectivity (low is best)

- **conductance**: fraction of total edge volume that points outside the cluster, $\frac{f_c}{2m_c+f_c}$
- **normalized cut**: $\frac{f_c}{2m_c+f_c} + \frac{f_c}{2(m-m_c)+f_c}$
- **Flake's out degree fraction**: fraction of nodes in C that have more edges pointing outside than inside

$$\frac{|\{u : u \in C \text{ and } |\{(u,v) \in E : v \in C\}| < k_u/2\}|}{n_c}$$

- Other: **maximum out degree fraction (odf)**, **average odf**.

Quality criteria V

(D) Based on a network model (high is best)

- **modularity**: difference between nr. of edges in C and the expected nr. of edges $E[m_c]$ of a random graph with the same degree distribution

$$\frac{1}{4m}(m_c - E[m_c])$$

Quality criteria VI

So far, we defined metrics for single communities. In order to measure them over the whole network, the usual approach is to compute a weighted average where weights are proportional to community volume, namely:

$$metric(G) = \sum_{C \in comm(G)} \frac{n_C}{n} * metric(C)$$

Contents

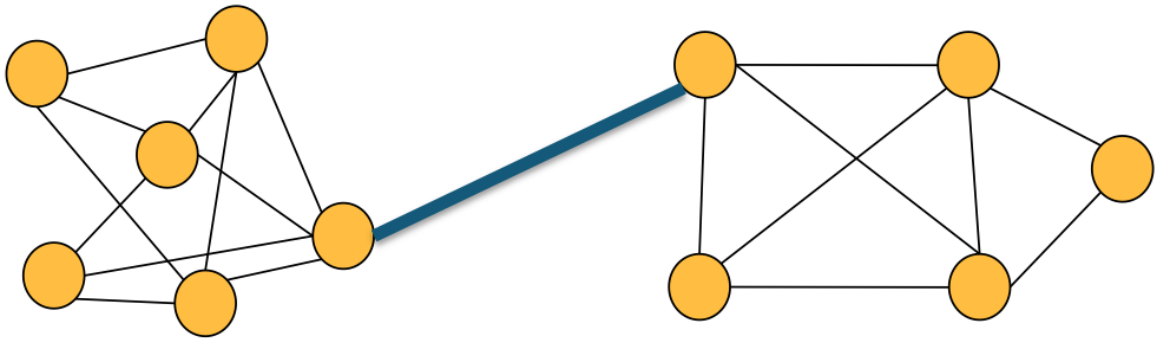
The Girvan-Newman algorithm

A *divisive* hierarchical algorithm [Girvan and Newman, 2002]

Edge betweenness

The betweenness of an edge is the nr. of shortest-paths in the network that pass through that edge

It uses the idea that "bridges" between communities must have high edge betweenness

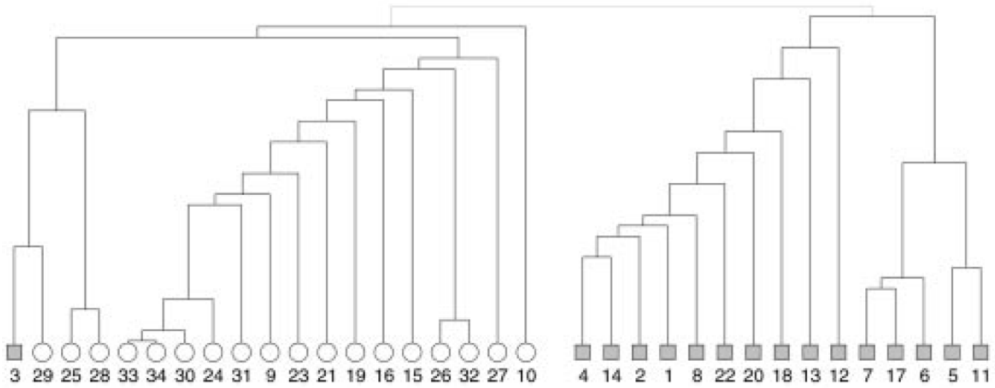


The Girvan-Newman algorithm

Pseudocode

1. Compute betweenness for all edges in the network
2. Remove the edge with highest betweenness
3. Go to step 1 until no edges left

Result is a dendogram



Definition of modularity [Newman, 2010]

Using a *null* model

Random graphs are not expected to have community structure, so we will use them as null models.

$$Q = (\text{nr. of intra-cluster communities}) - (\text{expected nr of edges})$$

In particular:

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - P_{ij})\ \delta(C_i, C_j)$$

where P_{ij} is the expected number of edges between nodes i and j under the null model, C_i is the community of vertex i, and $\delta(C_i, C_j) = 1$ if $C_i = C_j$ and 0 otherwise.

How do we compute P_{ij}?

Using the "configuration" null model

The "configuration" random graph model choses a graph with the same degree distribution as the original graph uniformly at random.

- Let us compute P_{ij}
- There are $2m$ *stubs* or half-edges available in the configuration model
- Let p_i be the probability of picking at random a stub incident with i

$$p_i = \frac{k_i}{2m}$$

- The probability of connecting i to j is then $p_i p_j = \frac{k_i k_j}{4m^2}$
- And so $P_{ij} = 2m p_i p_j = \frac{k_i k_j}{2m}$

Properties of modularity

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

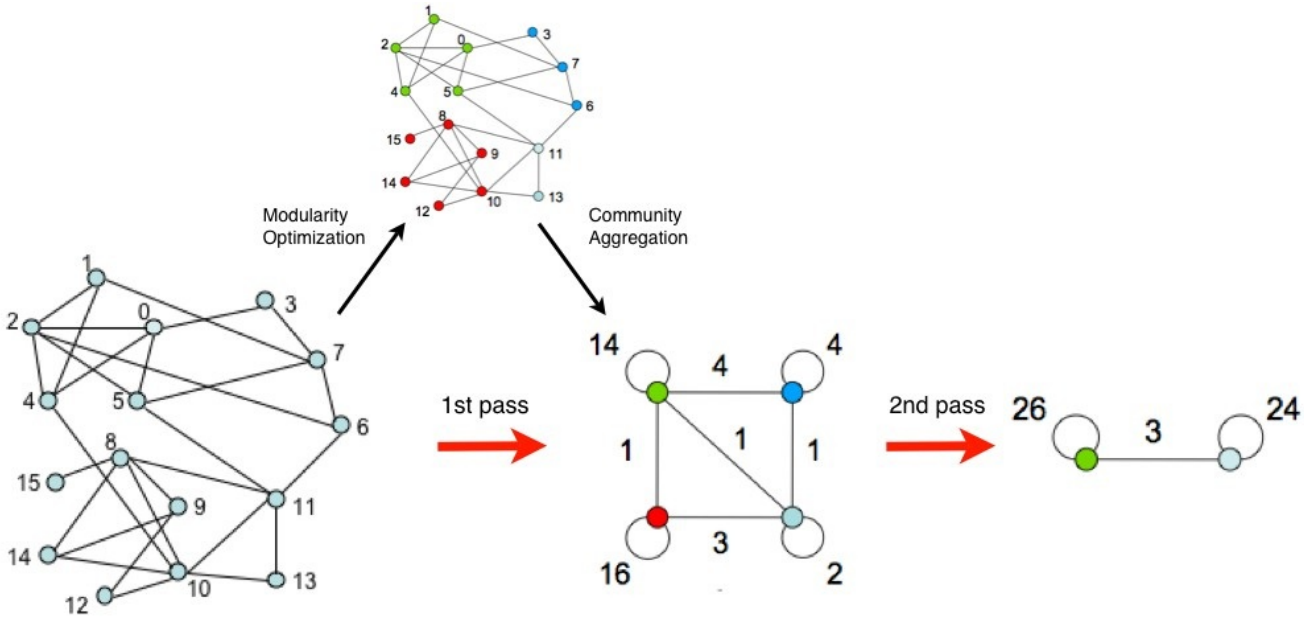
- Q depends on nodes in the same clusters only
- Larger modularity means better communities (better than random intra-cluster density)
- $Q \leq \frac{1}{2m} \sum_{ij} A_{ij} \ \delta(C_i, C_j) \leq \frac{1}{2m} \sum_{ij} A_{ij} \leq 1$
- Q may take negative values
  - partitions with large negative Q implies existence of cluster with small internal edge density and large inter-community edges

Algorithms to maximize modularity

- Greedy
  - Hierarchical: join clusters leading to largest increase in modularity [Newman, 2004]
  - Clauset algorithm: fast version using nice data structures that exploit sparsity [Clauset et al., 2004]
  - Louvain algorithm [Blondel et al., 2008]
- Spectral algorithms [Newman, 2006]
- .. and many others

The Louvain method [Blondel et al., 2008]

Considered state-of-the-art



Pseudocode

1. Repeat until local optimum reached
   - 1.1 Phase 1: partition network greedily using modularity
   - 1.2 Phase 2: agglomerate found clusters into new nodes

The Louvain method

Phase 1: optimizing modularity

Pseudocode for phase 1

1. Assign a different community to each node
2. For each node i
   - For each neighbor j of i, consider removing i from its community and placing it to j's community
   - Greedily chose to place i into community of neighbor that leads to highest modularity gain
3. Repeat until no improvement can be done

The Louvain method

Phase 2: agglomerating clusters to form new network

Pseudocode for phase 2

1. Let each community C_i form a new node i
2. Let the edges between new nodes i and j be the sum of edges between nodes in C_i and C_j in the previous graph (notice there are self-loops)

The Louvain method

Observarions

- The output is also a hierarchy
- Works for weighted graphs, and so modularity has to be generalized to

$$Q^w = \frac{1}{2W} \sum_{ij} \left(W_{ij} - \frac{s_i s_j}{2W} \right) \delta(C_i, C_j)$$

where W_{ij} is the weight of undirected edge (i, j), $W = \sum_{ij} W_{ij}$ and $s_i = \sum_k W_{ik}$.

Spectral modularity optimization [Newman, 2006] I

- Define modularity matrix **B** with elements $B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$
- Let **s** be a vector representing a partition of the nodes into two communities C_1 and C_2: $s_i = +1$ if $i \in C_1$, $s_i = -1$ if $i \in C_2$
- Then:

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

$$= \frac{1}{2m} \sum_{ij} B_{ij} \, \delta(C_i, C_j)$$

$$= \frac{1}{4m} \sum_{ij} B_{ij} \, (s_i s_j + 1)$$

Spectral modularity optimization [Newman, 2006] II

Equivalently, optimize

$$Q = \frac{1}{4m} \sum_{ij} B_{ij} s_i s_j = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s}$$

- Let $\{\mathbf{u}_k\}_k$ be eigenvectors of $\mathbf{B}$; since $\mathbf{B}$ is symmetric and real, they form a orthonormal basis ($\mathbf{u}_k^T \mathbf{u}_{k'} = 0$ for $k \neq k'$)

Spectral modularity optimization [Newman, 2006] III

- We can decompose vector $\mathbf{s} = \sum_k a_k \mathbf{u}_k$ such that $a_k = \mathbf{u}_k^T \mathbf{s}$

$$
\begin{aligned}
Q &= \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s} \\
&= \frac{1}{4m} (\sum_k a_k \mathbf{u}_k^T) \mathbf{B} (\sum_{k'} a_{k'} \mathbf{u}_{k'}) \\
&= \frac{1}{4m} (\sum_k a_k \mathbf{u}_k^T \beta_k) (\sum_{k'} a_{k'} \mathbf{u}_{k'}) \\
&= \frac{1}{4m} \sum_{k,k'} a_k a_{k'} \mathbf{u}_k^T \mathbf{u}_{k'} \beta_k \\
&= \frac{1}{4m} \sum_k a_k^2 \beta_k = \frac{1}{4m} \sum_k (\mathbf{u}_k^T \mathbf{s})^2 \beta_k
\end{aligned}
$$

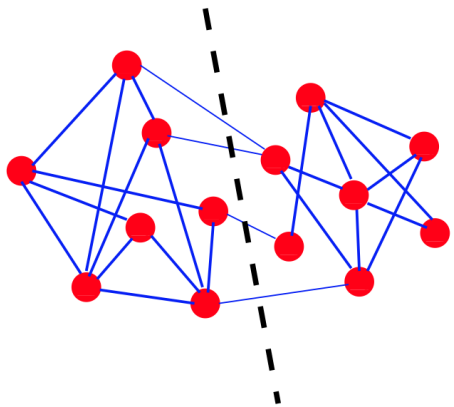
Spectral modularity optimization [Newman, 2006] IV

- In order to maximize $Q = \frac{1}{4m} \sum_k (\mathbf{u}_k^T \mathbf{s})^2 \beta_k$, we can look at the largest eigenvalue β_1 corresponding to eigenvector $\mathbf{u}_1$ and define $s_i = sign(\mathbf{u}_{1\,i})$

Graph partitioning algorithms

Divide the current network into groups of predefined size such that the number of edges between the groups is minimized

- The *minimum bisection problem*, is a special case that considers partitioning the network into two groups of equal size (NP-hard, of course)



- Then, in order to obtain a full partition one iteratively finds minimum bisections (not great for community detection)

Minimum bisection algorithms

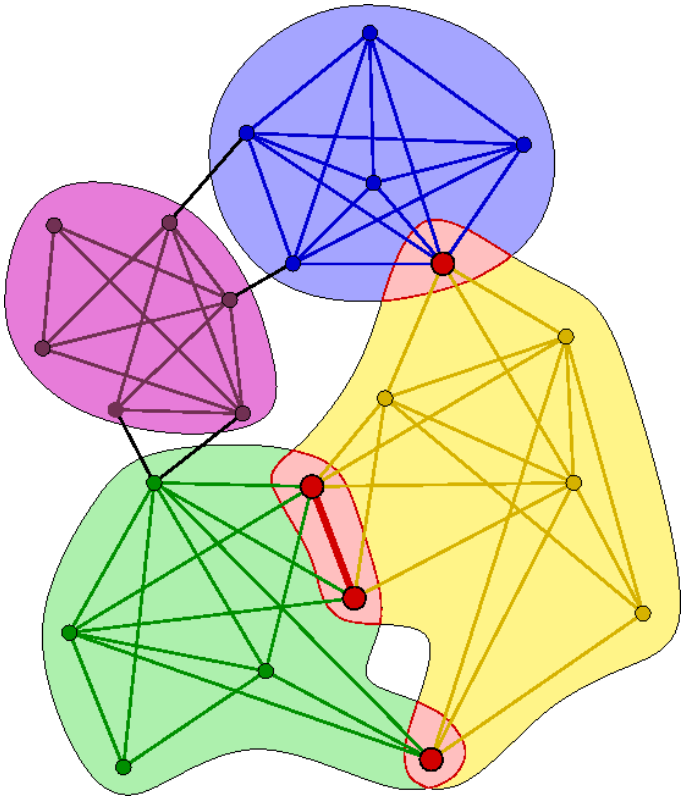
- Kernighan-Lin algorithm [Kernighan and Lin, 1970]
- Spectral bisection algorithm
- Conductance, cut ratio, normalized cut ration minimization procedures
- ..

Kernighan-Lin algorithm [Kernighan and Lin, 1970]

- Greedily optimize objective function: internal – external edges
- Start with random bisection
- Swap subsets of equal size on opposite ends to see whether objective function improves (could be singletons)
- typically used to improve an existing bisection

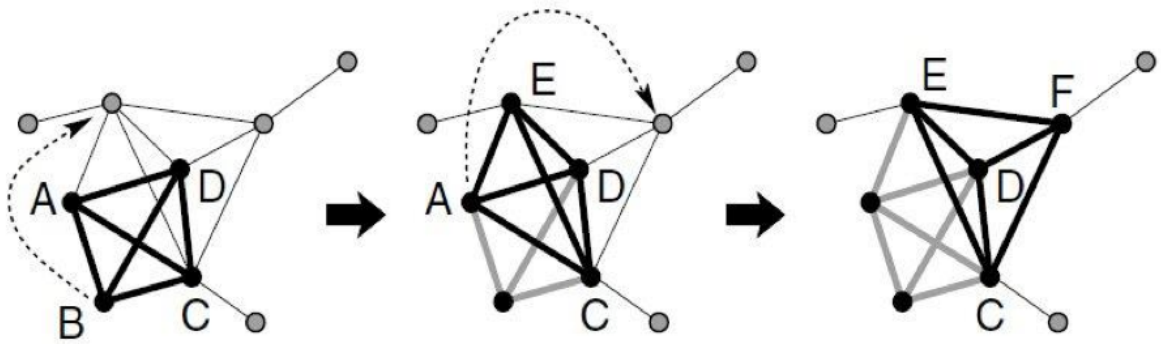
Clique percolation method [Palla et al., 2005]

Generates *overlapping* clusters!



Clique percolation method

- Detects densely connected communities
- k-clique: complete subgraph on k nodes
- Adjacent k-cliques: two k-cliques that share $k-1$ nodes
- Module: union of adjacent k-cliques



References I


Arratia, A. (2014).
Computational Finance. An Introductory Course with R.
Atlantis Press – Springer.


Blondel, V. D., Guillaume, J.-I., Lambiotte, R., and Lefebvre, E. (2008).
Fast unfolding of community hierarchies in large networks.
Networks, pages 1–6.


Clauset, A., Newman, M. E. J., and Moore, C. (2004).
Finding community structure in very large networks.
Physical Review E.

References II


Fortunato, S. (2010).
Community detection in graphs.
Physics Reports, 486(35):75 – 174.


Girvan, M. and Newman, M. E. J. (2002).
Community structure in social and biological networks.
Proceedings of the National Academy of Sciences of the United States of America, 99:7821–7826.


Kernighan, B. W. and Lin, S. (1970).
An efficient heuristic procedure for partitioning graphs.
Bell Sys. Tech. J., 49(2):291–308.

References III


Newman, M. (2010).
Networks: An Introduction.
Oxford University Press, USA, 2010 edition.


Newman, M. E. J. (2004).
Fast algorithm for detecting community structure in networks.
PHYSICAL REVIEW E, 69:1–5.


Newman, M. E. J. (2006).
Modularity and community structure in networks.
Proceedings of the National Academy of Sciences of the United States of America, 103:8577–8582.

References IV


Palla, G., Derényi, I., Farkas, I., and Vicsek, T. (2005).
Uncovering the overlapping community structure of complex networks in nature and society.
Nature, 435:814–818.


Yang, J. and Leskovec, J. (2012).
Defining and evaluating network communities based on ground-truth.
In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, pages 3:1–3:8, New York, NY, USA. ACM.