

Community structure in networks

Argimiro Arratia & Ramon Ferrer-i-Cancho

Universitat Politècnica de Catalunya

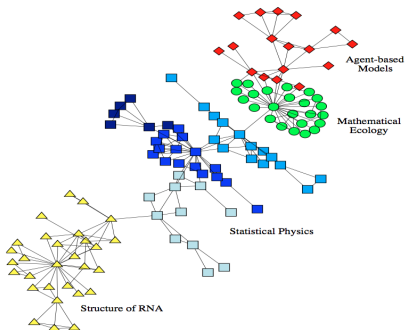
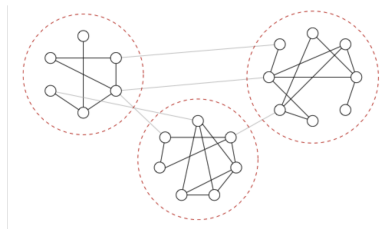
Complex and Social Networks (2024-2025)
Master in Innovation and Research in Informatics (MIRI)

Instructors

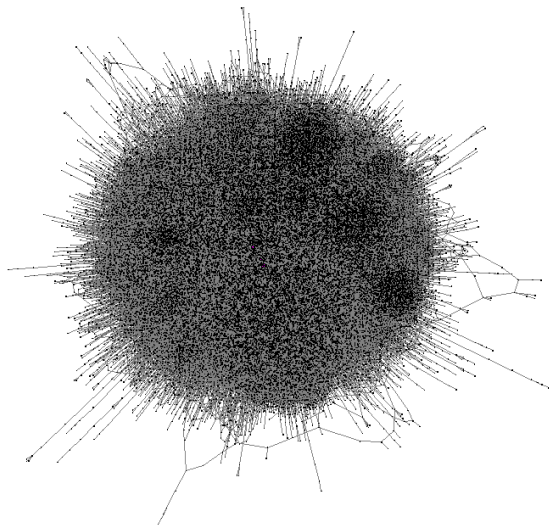
- ▶ Ramon Ferrer-i-Cancho, rferrericanco@cs.upc.edu,
<http://www.cs.upc.edu/~rferrericanco/>
- ▶ Argimiro Arratia, argimiro@cs.upc.edu,
<http://www.cs.upc.edu/~argimiro/>

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

What is community structure?

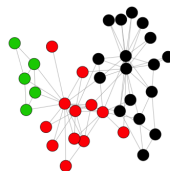
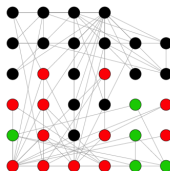
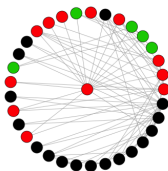
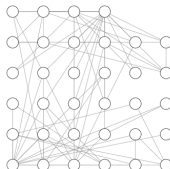
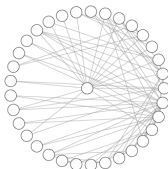


Why is community structure important?



.. but don't trust visual perception

it is best to use objective algorithms



Outline for today

Clustering algorithms (General outlook)

- Hierarchical clustering algorithms

Quantifying the quality of community structure

[Yang and Leskovec, 2012, Arratia and Renedo-Mirambell, 2021]

- Quality measures for weighted graphs

- Significance and Stability

Back to methods for detection of community structure

[Fortunato, 2010]

- Girvan-Newman algorithm

- Modularity optimization algorithms

 - Louvain

- Graph partitioning algorithms

Contents

Clustering algorithms (General outlook)

Hierarchical clustering algorithms

Quantifying the quality of community structure

[Yang and Leskovec, 2012, Arratia and Renedo-Mirambell, 2021]

Quality measures for weighted graphs

Significance and Stability

Back to methods for detection of community structure

[Fortunato, 2010]

Girvan-Newman algorithm

Modularity optimization algorithms

Graph partitioning algorithms

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 $d(x, x) = 0$
- ▶ $d(x, y) = d(y, x)$
- ▶ $d(x, y) \leq d(x, z) + d(z, y)$ (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 we want to interpret high value of similarity as high similarity, and we are working with distance metric $d(x, y)$, then consider its inverse: $s(x, y) = 1/d(x, y)$ or $1/d(x, y) + 0.5$.

NB: We are here concerned 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 \mathbf{A} be the adjacency matrix of the network, i.e. $A_{ij} = 1$ if $(i, j) \in E$ and 0 otherwise.

► **Jaccard index:**

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

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

Similarity measures w_{ij} for nodes II

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

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

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**

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

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

¹Uses 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".

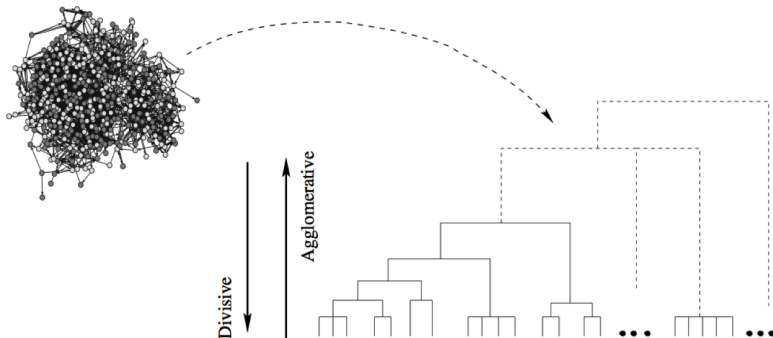
In math:

$$\begin{aligned}\Delta(X, Y) &= \sum_{i \in X \cup Y} \|z_i - c_{X \cup Y}\|^2 - \sum_{i \in X} \|z_i - c_X\|^2 - \sum_{i \in Y} \|z_i - c_Y\|^2 \\ &= \frac{|X| \times |Y|}{|X| + |Y|} \|c_X - c_Y\|^2\end{aligned}$$

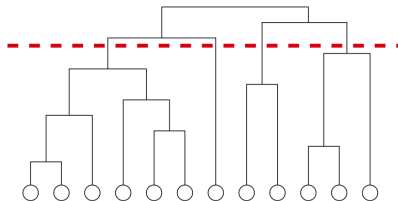
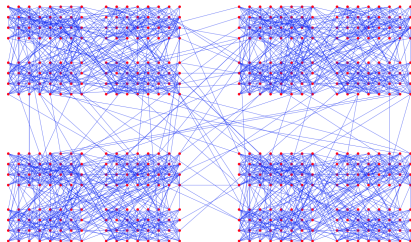
- ▶ single linkage : tends to make too small (in size) clusters
- ▶ complete: too big and fewer clusters
- ▶ average : more or less regular
- ▶ Ward's : tends to minimise the total within cluster variance

Hierarchical clustering

From hairball to *dendrogram*



Suitable if input network has hierarchical structure



Agglomerative hierarchical clustering [Newman, 2010]

Ingredients

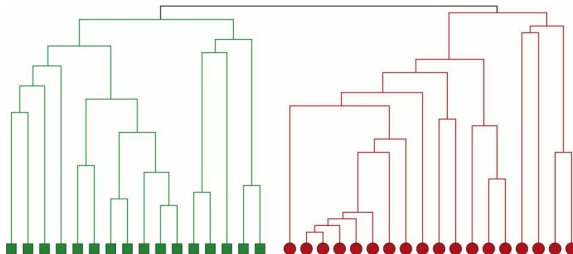
- ▶ Similarity measure between nodes
- ▶ Similarity measure between *sets of nodes*

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]

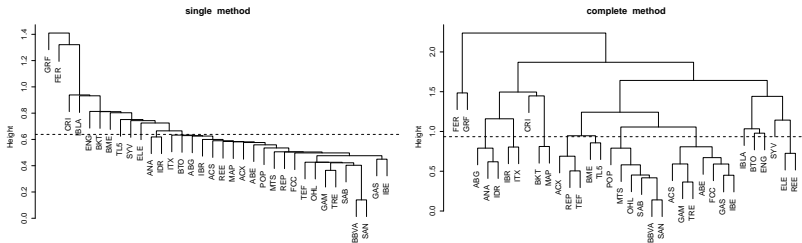


Figure: Dendrograms for single and complete inter-cluster linkages and dissimilarity measure $2(1 - \rho(\mathbf{x}, \mathbf{y}))$.

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

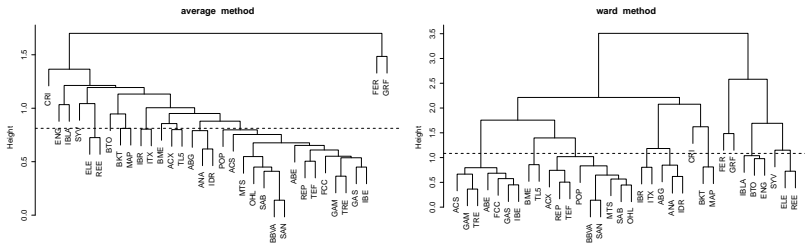


Figure: Dendrograms for average and Ward inter-cluster linkages and dissimilarity $2(1 - \rho(\mathbf{x}, \mathbf{y}))$.

Contents

Clustering algorithms (General outlook)

Hierarchical clustering algorithms

Quantifying the quality of community structure

[Yang and Leskovec, 2012, Arratia and Renedo-Mirambell, 2021]

Quality measures for weighted graphs

Significance and Stability

Back to methods for detection of community structure

[Fortunato, 2010]

Girvan-Newman algorithm

Modularity optimization algorithms

Graph partitioning algorithms

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.

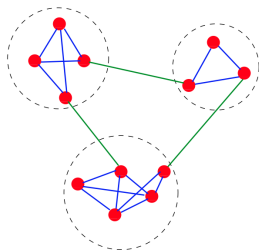
$$\delta(G) = \frac{\text{nr. edges in } G}{n(n - 1)/2}$$

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 (aka Clustering coef., transitivity)**: 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

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

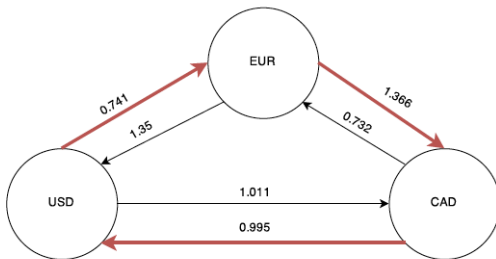
Quality criteria VI

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:

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

Quality measures for weighted networks I

What if our network has weights on edges?



$G(V, E)$ be an undirected graph of order $n = |V|$ and size $m = |E|$. In the case of a weighted graph $\tilde{G}(V, \tilde{E})$, we use $\tilde{m} = \sum_{e \in \tilde{E}} w(e)$ the sum of all edge weights.

¹For every variable or function defined over the unweighted graph, we use a “ \sim ” to denote its weighted counterpart

Quality measures for weighted networks I

Given $S \subset V$, let $n_S = |S|$, $m_S = |\{(u, v) \in E : u \in S, v \in S\}|$, and in the weighted case $\tilde{m}_S = \sum_{(u,v) \in E: u,v \in S} w_{uv}$.
(We use w_{uv} instead of $w((u, v))$.)

- ▶ $c_S = |\{(u, v) \in E : u \in S, v \notin S\}|$, number of edges connecting S to the rest of the graph.
- ▶ $\tilde{c}_S = \sum_{(u,v) \in E: u \in S, v \notin S} w_{uv}$; the sum of weights of all edges connecting S to $G \setminus S$.
- ▶ $\tilde{d}(u) = \sum_{v \neq u} w_{uv}$ is the extension of vertex degree $d(u)$ to weighted graphs; the sum of weights of edges incident to u .
- ▶ $d_S(u) = |\{v \in S : (u, v) \in E\}|$ and $\tilde{d}_S(u) = \sum_{v \in S} w_{uv}$ are the (unweighted and weighted, respectively) degrees restricted to the subgraph S .

Quality measures for weighted networks II

	unweighted $f(S)$	weighted $f(S)$
↑ Internal density	$\frac{m_S}{n_S(n_S-1)/2}$	$\frac{\tilde{m}_S}{n_S(n_S-1)/2}$
↑ Edges Inside	m_S	\tilde{m}_S
↑ Average Degree	$\frac{2m_S}{n_S}$	$\frac{2\tilde{m}_S}{n_S}$
↓ Expansion	$\frac{c_S}{n_S}$	$\frac{\tilde{c}_S}{n_S}$
↓ Cut Ratio	$\frac{c_S}{n_S(n - n_S)}$	$\frac{\tilde{c}_S}{n_S(n - n_S)}$
↓ Conductance	$\frac{c_S}{2m_S + c_S}$	$\frac{\tilde{c}_S}{2\tilde{m}_S + \tilde{c}_S}$
↓ Normalized Cut	$\frac{c_S}{2m_S + c_S} + \frac{c_S}{2(m - m_S) + c_S}$	$\frac{\tilde{c}_S}{2\tilde{m}_S + \tilde{c}_S} + \frac{\tilde{c}_S}{2(\tilde{m} - \tilde{m}_S) + \tilde{c}_S}$
↓ Maximum ODF	$\max_{u \in S} \frac{ \{(u, v) \in E: v \notin S\} }{d(u)}$	$\max_{u \in S} \frac{\sum_{v \notin S} w_{uv}}{\tilde{d}(u)}$
↓ Average ODF	$\frac{1}{n_S} \sum_{u \in S} \frac{ \{(u, v) \in E: v \notin S\} }{d(u)}$	$\frac{1}{n_S} \sum_{u \in S} \frac{\sum_{v \notin S} w_{uv}}{\tilde{d}(u)}$

Community scoring functions $f(S)$ for unweighted and weighted networks.

Quality measures for weighted networks III

Clustering coefficient

Assuming weights in the interval $[0, 1]$.

- ▶ For $t \in [0, 1]$ let A_t be the adjacency matrix with elements $A_{ij}^t = 1$ if $w_{ij} \geq t$ and 0 otherwise.
- ▶ Let C_t the clustering coefficient of the graph defined by A_t .
- ▶ The resulting weighted clustering coefficient is defined as

$$\tilde{C} = \int_0^1 C_t dt \quad (1)$$

Quality measures for weighted networks IV

For networks where the weights are either not bounded or bounded in an interval not $[0, 1]$, simply take

$$\tilde{C} = \frac{1}{\bar{w}} \int_0^{\bar{w}} C_t dt, \quad (2)$$

where \bar{w} can be either the upper bound or, in the case of networks with no natural bound, the maximum edge weight. (The integral is computed as a sum of the values of C_t .)

Significance and Stability²

Significance The scoring functions (for unweighted or weighted graphs) provide measures of clustering *significance*. A partition of a network into clusters is *significant* if it obtains better scores than those for a comparable network with no community structure. This reference network is obtained by rewiring edges (or transferring weights) while keeping degree distribution constant, to get a comparable network with uniformly distributed edges.

Stability This means how much a clustering remains unchanged under small perturbations of the network. In the case of weighted networks, these could include the addition and removal of vertices, as well as the perturbation of edge weights.

²[Arratia and Renedo-Mirambell, 2021]

More on Stability

The idea is that meaningful clusters should capture an inherent structure in the data and not be overly sensitive to small and/or local variations, or the particularities of the clustering algorithm.

Procedure

(repeat 100 times)

- ▶ a bootstrap resampling is done on input graph (i.e. uniform sampling of vertices with replacement),
- ▶ selected clustering algorithms are applied,
- ▶ for each app. measure the variation of the communities obtained in resampled graphs w.r.to original

Report mean similarity clustering measure for each algorithm.

Some measures of (global) clustering similarity: Variation of Information, Reduced Mutual Information, Rand Index.

Clustering similarity measures I

All are based on the *contingency table of the labellings*.

Consider two labelings (or partitions) of n elements, $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_R\}$ and $\mathcal{P}' = \{\mathcal{P}'_1, \dots, \mathcal{P}'_S\}$. Define

$$a_r = |\mathcal{P}_r| = \sum_{s=1}^S c_{rs} \quad \# \text{ elements label } r \text{ in 1st part.}$$

$$b_s = |\mathcal{P}'_s| = \sum_{r=1}^R c_{rs} = \# \text{ elements label } s \text{ in 2nd part.}$$

$$c_{rs} = |\mathcal{P}_r \cap \mathcal{P}'_s| = \# \text{ elements label } r \text{ in 1st part. and label } s \text{ in 2nd.}$$

Define the probability $P(r)$ (respectively, $P(s)$) of an object chosen uniformly at random has label r (resp. s), and the probability $P(r, s)$ that it has both labels r and s , that is

$$P(r) = \frac{a_r}{n}, \quad P(s) = \frac{b_s}{n}, \quad P(r, s) = \frac{c_{rs}}{n} \quad (3)$$

Clustering similarity measures II

	\mathcal{P}'_1	\mathcal{P}'_2	...	\mathcal{P}'_S	sum
\mathcal{P}_1	c_{11}	c_{12}	...	c_{1S}	a_1
\mathcal{P}_2	c_{21}	c_{22}		c_{2S}	a_2
	
\mathcal{P}_R	c_{R1}	c_{R2}	...	c_{RS}	a_r
sum	b_1	b_2	...	b_S	$n = \sum c_{ij}$

Table: Contingency table of partitions \mathcal{P} and \mathcal{P}' , with labelings r and s .

Variation of Information

The VI between two clusterings [Meilă, 2007], is defined as

- ▶ The entropy of a partition $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_R\}$ of a set is :

$$\mathcal{H}(r) = - \sum_{r=1}^R P(r) \log(P(r)), \quad (4)$$

- ▶ The mutual information is defined as:

$$I(r; s) = \sum_{r=1}^R \sum_{s=1}^S P(r, s) \log \left(\frac{P(r, s)}{P(r)P(s)} \right) \quad (5)$$

- ▶ The variation of information of partitions \mathcal{P} and \mathcal{P}' is given by:

$$VI(r; s) = \mathcal{H}(r) + \mathcal{H}(s) - 2I(r; s) \quad (6)$$

The R package clustAnalytics [Renedo-Mirambell, 2022]

clustAnalytics: Cluster Evaluation on Graphs

- ▶ Evaluates the stability and significance of clusters on 'igraph' graphs.
- ▶ Supports weighted and unweighted graphs.
- ▶ Implements the cluster evaluation methods defined by [Arratia and Renedo-Mirambell, 2021].
- ▶ includes an implementation of the Reduced Mutual Information
- ▶ includes methods to synthetically generate a weighted network with a ground truth community structure, and binomial or scale-free degree distribution.

Contents

Clustering algorithms (General outlook)

Hierarchical clustering algorithms

Quantifying the quality of community structure

[Yang and Leskovec, 2012, Arratia and Renedo-Mirambell, 2021]

Quality measures for weighted graphs

Significance and Stability

Back to methods for detection of community structure

[Fortunato, 2010]

Girvan-Newman algorithm

Modularity optimization algorithms

Graph partitioning algorithms

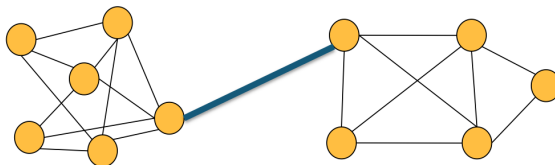
The Girvan-Newman algorithm

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

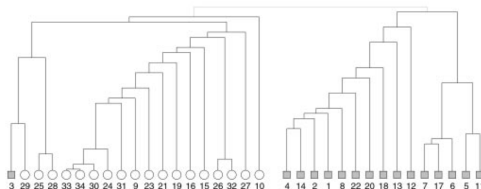


The Girvan-Newman algorithm

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 dendrogram



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

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

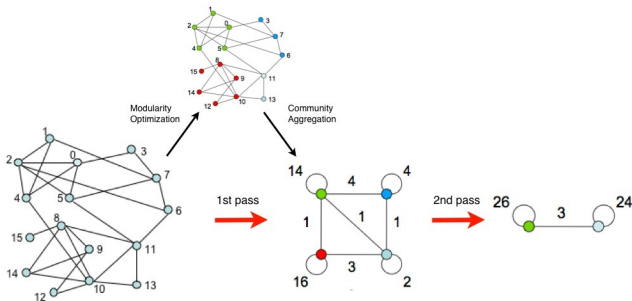
- ▶ Q depends on nodes in the same clusters only
- ▶ Larger modularity means better communities (better than random intra-cluster density)
- ▶ $Q \leq \frac{1}{2m} \sum_{ij} A_{ij} \delta(C_i, C_j) \leq \frac{1}{2m} \sum_{ij} A_{ij} \leq 1$
- ▶ Q may take negative values
 - ▶ partitions with large negative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

Observations

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

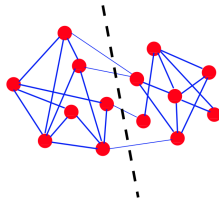
$$Q^w = \frac{1}{2W} \sum_{ij} \left(W_{ij} - \frac{s_i s_j}{2W} \right) \delta(C_i, C_j)$$

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

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
- ▶ ..

References I



Arratia, A. (2014).

Computational Finance. An Introductory Course with R.
Atlantis Press – Springer.



Arratia, A. and Renedo-Mirambell, M. (2021).

Clustering assessment in weighted networks.
PeerJ Computer Science, 7:e600.



Blondel, V. D., Guillaume, J.-l., Lambiotte, R., and Lefebvre, E. (2008).

Fast unfolding of community hierarchies in large networks.
Networks, pages 1–6.

References II



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



Fortunato, S. (2010).
Community detection in graphs.
Physics Reports, 486(3–5):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.

References III



Kernighan, B. W. and Lin, S. (1970).

An efficient heuristic procedure for partitioning graphs.
Bell Sys. Tech. J., 49(2):291–308.



Meilă, M. (2007).

Comparing clusterings - an information based distance.
Journal of Multivariate Analysis, 98(5):873 – 895.



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.

References IV



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.



Renedo-Mirambell, M. (2022).

clustAnalytics: Cluster Evaluation on Graphs (R package v 0.5).

<https://CRAN.R-project.org/package=clustAnalytics>.

References V



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.