Community Structure

Blocks and Communities: Definition
The general idea of blocks and communities is that nodes of a network can be grouped together in homogeneous sets, based on the network topology. The problem of automatically discovering those groups is one of the most studied problem of network science, but also one of the most difficult to properly define.

Partitions/Overlap
We must differentiate two types of node grouping:

1. A Partition of a graph is a division of its nodes such as each group of them belongs to one single group.
2. Overlapping communities/ blocks allow, on the contrary, nodes belonging to several groups. Unless specified differently, they also allow nodes to belong to no group.

Algorithms searching partitions are much more common than those searching for overlapping groups, due to the increased complexity of the later task. Overlapping community detection is, nevertheless, an active field of research.

Community structure
The idea of having a network structured in communities is defined as an analogy with communities in social networks. Communities are therefore defined (informally) as groups of nodes that are strongly connected between themselves (high internal density) and more weakly connected to the rest of the network (low external density).

This definition however cannot be translated unambiguously into a mathematical formulation. The problem of community detection, or community discovery, is therefore complex to define.

Modularity: resolution limit
It is important to remember that the Modularity is (only a) quality function, not a definition of the quality of communities. An important drawback of Modularity is known as the limit of resolution: it says that partitions of maximal modularity are biased toward a particular scale, i.e., for a graph of a given size (#nodes, #edges), communities smaller or larger than a certain size cannot be found. The typical example of this limit is the clique-ring structure (set of cliques connected by a single edge), in which the expected partition is to have one community by clique, while the solution of highest modularity puts several cliques in the same community, when we increase the number of cliques.

Modularity: null model
The modularity as expressed above compares the number of edges inside communities to the expected number of edges in a null model, i.e., a randomized version of the graph. In the original version, this null model is the configuration model (as easily recognized in the factor \( \frac{1}{2L} \)). Variants of the modularity have been proposed using different null models, for instance an ER null model, or a gravity model to take into account the effect of geographic distance.

Multi-resolution Modularity
A simple solution has been proposed to the limit of resolution, consisting in adding a resolution parameter \( \lambda \) to tune the desired resolution. It says that partitions of maximal modularity are biased toward a scale \( \lambda \), i.e., for a graph of a given size (#nodes, #edges), communities smaller or larger than a certain size cannot be found. The typical example of this limit is the clique-ring structure (set of cliques connected by a single edge), in which the expected partition is to have one community by clique, while the solution of highest modularity puts several cliques in the same community, when we increase the number of cliques.

Definition
\[ C = \text{a community partition, or, more generally, a set of sets of nodes} \]
\[ c_i = \text{community } i, \text{ a set of nodes} \]

Modularity
The most famous quality function to measure the quality of partitions is called the Modularity. Introduced in it is defined for a partition \( C \) and a graph \( G \) as the difference between the fraction of observed internal edges and the expected fraction of internal edges if \( G \) were rewired according to a configuration model, i.e., preserving the degrees of nodes. More formally,

\[ Q = \frac{1}{2L} \sum_{i=1}^{C} (L_{i} - \frac{1}{2}K_{i}^2) \]

with \( L_{i} = L(H(c_i)) \) the number of edges inside community \( i \) and \( K_{i} = \sum_{u \in c_i} k_u \) the sum of degrees of nodes in community \( i \). The original formulation of modularity, often found in the literature, is:

\[ Q = \frac{1}{2L} \sum_{uv} A_{uv} - \frac{k_u k_v}{2L} \delta(c_u, c_v) \]

with \( \delta(c_u, c_v) \) the kronecker delta between communities, i.e., \( \delta(c_u, c_v) = 1 \) if nodes \( u \) and \( v \) belongs to the same community, \( 0 \) otherwise.

\[ ^{a}\text{Girvan and Newman 2002} \]

\[ ^{b}\text{Fortunato and Barthelemy 2007} \]

\[ ^{c}\text{Reichardt and Bornholdt 2006} \]
**Mod. maximization: Girvan Newman**

Several of the most popular community detection algorithms have as objective to discover the partition of highest modularity. This is a difficult problem, and thus existing approaches are based on heuristics.

The original method by Girvan and Newman first builds a dendrogram by iteratively removing the edge of highest betweenness. It is called a *divise* approach: At the top of the dendrogram, there is a single community, then 2, 3, 4 etc., until each node is in its own community. Modularity is used as a criterium to cut the dendrogram.

![Dendrogram](Image)

**Mod. maximization: Louvain method**

The **Louvain method** is certainly the most used method for community detection. Its objective is to optimize the modularity using a greedy, agglomerative approach:

**Step 1** Optimizing modularity at a hierarchical level
- Each node starts in its own community
- Repeat until convergence:
  1. FOR each node, compute the gain in modularity of adding it to the community of each of its neighbors
  2. choose the decision that increases the most the modularity (the best decision can be to keep the node in the same community)

**Step 2** Global algorithm
- Repeat until convergence:
  1. Optimize modularity for the current hierarchical level according to Step 1
  2. Move to a higher hierarchical level by computing an *induced network*: each community becomes a node, the weight of the edge between nodes $i$ and $j$ corresponds to the number of edges between nodes of $c_i$ and nodes of $c_j$.

The result of Louvain algorithm is therefore a hierarchy of communities.

![Louvain Algorithm](Image)

**Louvain method strengths and weaknesses**

The main reason explaining the popularity of the Louvain method to this day is its scalability. The algorithm is very scalable in practice on real graphs, for several reasons: 1) it is a greedy approach, 2) By checking only the interest of moving to neighbor’s communities, it benefits from the sparsity of networks, 3) Modularity gains of a partition change can be computed locally, using its definition as a sum of independent values for each community.

Another advantage of the Louvain method is that results at lower hierarchical levels can naturally mitigate the problem of the resolution limit, for instance on the ring clique example, Louvain finds each clique in its community at the first level, and only in a second level yield the problematic partition. However, it has also be shown that the greedy nature of the algorithm could lead to having counter-intuitive structures, such as disconnected communities. The authors thus introduced a variant of the algorithm called Leiden, solving this problem.

![Louvain strength and weakness](Image)

**Infomap**

Infomap is a method based on an objective function different from the Modularity. Its objective is to minimize the description of an average random walk in the network, i.e. maximize the compression of the description of such a walk. More formally, the code length to minimize for partition $C$ is described as:

$$H(C) = qH(\forall) + \sum_i p_i H(C_i)$$

with $q$ the probability for a move to be between modules, $H(\forall)$ the information required to encode a move between modules, $p_i$ the probability for a move to be inside community $i$ and $H(C_i)$ the information required to encode a move inside community $i$.

A greedy optimization algorithm, similar in nature to the one of Louvain, is then used to minimize this description length. Compared with Modularity, the main advantage of this approach is that it does not find communities in random networks. It is known also to suffer from a resolution limit, although not exactly similar to the one of Modularity.

**Stochastic Block Models (SBM)**

A stochastic block model is a random graph model defined by:

- $k$ number of blocks
- $b$ $n \times 1$ vector such as $b_i$ describes the index of the block of node $i$.
- $E$ $k \times k$ stochastic block matrix, such as $E_{ij}$ gives the number of edges between blocks $i$ and $j$ (or the probability to observe an edge between any pair of nodes chosen with one node in each of the two blocks).

![Stochastic Block Models](Image)
The objective of a community/block detection algorithm based on the SBM principle is thus to perform SBM inference, i.e., to find the parameters of the SBM that best explain the observed graph, usually in terms of maximizing the likelihood. Said differently, we search –among a certain class of models– the model that has the highest probability to generate the observed graph. Note that for an observed graph, for each partition in blocks $b$, there is a single block matrix $E$ that is relevant to consider, that can be found simply by counting the number of edges actually present between blocks in the graph.

More formally, the objective is:

$$b := \arg\max_b P(A|b)$$

Note that with this formulation, it is not possible to infer the number of clusters $k$, since the trivial solution in which each node belongs to its own block, with $E = A$ has a maximal probability to generate the observed graph. The desired number of clusters is thus a necessary parameter of this type of SBM inference.

Recently, new approaches have been proposed to infer also automatically the number of blocks. They adopt an approach from Information Theory called the Minimum Description Length (MDL), whose principle is to find the description which reduces the total cost of describing a graph, by minimizing both 1) the quantity of information needed to encode the graph, knowing that it is generated by a given model, and 2) the quantity of information needed to encode the model itself. Intuitively, a model with few blocks requires little information to be described, contrary to a model with many blocks. But a model with many blocks is more constrained, the graphs it generates are more specific, and therefore can be described at a lesser cost, knowing the model.

More formally, we can decompose the probability of observing a graph and a model as:

$$P(A, k, e, b) = P(A|k, e, b)P(k|e, b)P(e|b)P(b)$$

with the last three probabilities being priors. Said differently, we can define the number of bits required to encode a model as $L = - \log_2 P(k, b)$, the number of bits necessary to encode a graph knowing the model as $S = - \log_2 P(A|k, b)$ and thus the total cost to minimize as $S + L$.

The objective thus becomes:

$$b := \arg\min_b -\log_2 P(k, b) - \log_2 P(A|k, b)$$

Group inference using SBM is a very active field of research, and many variants have been proposed, including degree-corrected, nested, Overlapping, Mixed membership SBM, etc.

An introduction to the state of the art can be found for instance in.

A python library exists to apply recent methods to observed graphs.

Lee and Wilkinson 2019
https://graph-tool.skewed.de

Since there isn’t a unique accepted definition of what are good communities, the evaluation of the quality of a partition or set of communities is not a trivial task. There are two main approaches:

- Internal evaluation consists in using quality functions (e.g., Modularity) to give a score for a pair partition/graph
- External evaluation consists in comparing a computer partition to a ground truth reference partition.

Internal evaluation can be used to evaluate the quality of communities found on a network of interest, while External evaluation is mostly used to assess the quality of algorithms on benchmarks, to evaluate their performance before using them on a real network.

Several quality functions exist to evaluate the quality of a community partition of a graph. They can therefore be understood as different definitions of community structures. While some methods try directly to optimize one of those quality functions, some other methods are based on different principles (e.g., clique-based communities, consensus reaching based on game-theory, etc.). Quality functions can therefore be used a posteriori to assess the quality of communities they found. The most popular are:

- Modularity
- Information compression, as in Infomap or SBM
- Surprisal evaluates the departure of the observed partition from the expected distribution of nodes and links into communities given a null model, and is therefore related to Modularity

Partitions obtained by a given method can be compared with a ground truth. This approach is used on real networks, with a ground truth coming from metadata (e.g., classes in a network of social interactions between students), and on synthetic networks, with communities known by construction.

Although this is still discussed in the literature, it is mostly accepted that the evaluation on real networks using this approach is problematic because there is no guarantee that the labels used as ground truth are indeed related to the topological structure of the network, which is what communities are about.

Most popular methods for partitions comparisons are:

- NMI. Normalized Mutual Information, and its adjusted for chance variant, AMI.
- ARI. Adjusted Rand Index

But more generally, any method for cluster comparison can be used.

Peel, Larremore, and Clauset 2017
Dao, Bothorel, and Lenca 2020

Some quality functions are defined at the level of individual communities, instead of having one score for the whole partition. Those individual scores can however be combined to provide a global score, for instance using a weighted average. Some of the most popular are:

- Conductance, the fraction of all stubs of nodes in the community that points outside of it
- ODF. Out Degree Fraction, the average for every node of its fraction of neighbors inside the community
- Internal Transitivity, the clustering coefficient inside the community
- Scaled density, the ratio of the node density to the total graph density

Aldecoa and Marin 2013
Lee and Wilkinson 2019
https://graph-tool.skewed.de

Variants of the SBM

Internal Evaluation - Global

Internal Evaluation - By community

Evaluation of Community structures

External Evaluation

SBM infering the number of blocks

SBM inference

a Peixoto 2019
b https://graph-tool.skewed.de

c Aldecoa and Marin 2013

\(^{a}\)Lee and Wilkinson 2019
\(^{b}\)https://graph-tool.skewed.de

\(^{a}\)Lee and Wilkinson 2019
\(^{b}\)https://graph-tool.skewed.de

\(^{a}\)Aldecoa and Marin 2013
\(^{b}\)Lee and Wilkinson 2019
Overlapping communities

For many types of networks, the real organization of networks is thought to be overlapping, i.e., each node can belong to several communities. Think for instance of your personal social network: some of your family members might also be part of a group of friends, or some of your friends from high school might also be part of your friends from university, which are otherwise distinct groups. Detecting overlapping clusters is considered harder than non-overlapping ones, for at least two reasons: the search space (number of possible solutions) is much larger (and even infinite), and defining what good communities are is even harder, since there isn’t the natural limit for each edge to be either internal or external. A large number of methods have nevertheless been proposed. Extensions of non-overlapping quality functions have been proposed, such as the overlapping Modularity or overlapping NMI, but they are not as widely used as their more constrained counterparts.

Other meso-scale structures

Beyond the usual community structure, other types of network structural organizations have been proposed and studied. Some of the most widely known are:

- **Link communities**, in which communities are defined as sets of links. Searching for (non-overlapping) partitions of edges yield a structure in which nodes naturally belong to several groups, i.e., a community can corresponds to familial edges, another to professional edges, etc. (Ahn, Bagrow, and Lehmann 2010)

- **Fuzzy communities**, in which nodes belong to (often several) communities with a certain probability or strength (Liu 2018)

- **Core-Periphery structure**, already defined when we introduced the notion of k-cores

- **Nestedness**, corresponding to a network with a hierarchical organization such as elements with few connections tends to be connected to a subset of the neighbors of a parent node. (Pawar 2018)

- **Spatial organization**, in which the probability of observing an edge between nodes depends on their distance. (Barthélemy 2011)

Meso-scale organization

Beyond the usual community structure, other types of network structural organizations have been proposed and studied. Some of the most widely known are:

- **Link communities**, in which communities are defined as sets of links. Searching for (non-overlapping) partitions of edges yield a structure in which nodes naturally belong to several groups, i.e., a community can corresponds to familial edges, another to professional edges, etc. (Ahn, Bagrow, and Lehmann 2010)

- **Fuzzy communities**, in which nodes belong to (often several) communities with a certain probability or strength (Liu 2018)

- **Core-Periphery structure**, already defined when we introduced the notion of k-cores

- **Nestedness**, corresponding to a network with a hierarchical organization such as elements with few connections tends to be connected to a subset of the neighbors of a parent node. (Pawar 2018)

- **Spatial organization**, in which the probability of observing an edge between nodes depends on their distance. (Barthélemy 2011)

Going Further

Surveys: (Fortunato 2010) (Fortunato and Hric 2016)
On community detection approaches: (Rosvall, Delvenne, et al. 2019)
On stochastic Block Models: (Funke and Becker 2019)
Survey overlapping communities: (Xie, Kelley, and Szymanski 2013)
Community detection in dynamic networks: (Rossetti and Cazabet 2018)
On ground Truth and community detection: (Peel, Larremore, and Clauset 2017)
Community detection in neuroscience: (Betzel 2020)

References


