# Graph mining
# SD212
# Clustering

Thomas Bonald

2017 – 2018

These lecture notes introduce some metrics and algorithms for graph clustering, a key technique in graph analysis, also known as community detection. We refer to [2] for a survey on this topic.

## 1 Notion of clustering

Consider an undirected graph $G = (V, E)$ of $n$ nodes and $m$ edges, with $V = \{1, \dots, n\}$. Unless otherwise specified, we assume that there are no self-loops and no weights. We denote by $A$ the adjacency matrix and by $d_i = \sum_{j \in V} A_{ij}$ the degree of node $i$.

We are interested in partitioning the set of nodes $V$ into subsets called clusters so that "close" nodes (either neighbors or connected through many other nodes) tend to be in the same cluster. Formally, a clustering of the graph into $K$ clusters is a surjective function $C : V \to \{1, \dots, K\}$. We refer to $C^{-1}(k)$ as cluster $k$, for each $k = 1, \dots, K$. In general, the parameter $K$ is not given (unlike $K$-means for vector data in some Euclidian space) and one looks for the best clustering $C$ for any value of $K$.

## 2 Modularity

We need a metric to assess the quality of a clustering $C$. The usual metric is the modularity, defined by:

$$Q(C) = \frac{1}{2m} \sum_{i,j \in V} \left( A_{ij} - \frac{d_i d_j}{2m} \right) \delta_{C(i),C(j)}.$$

Observe that $Q(C) \in [-1, 1]$. The higher the modularity $Q(C)$, the better the clustering $C$. A graph with some clustering $C$ such that $Q(C) > 0$ is called *modular*.

The modularity is the difference between two terms. The first term corresponds to the proportion of edges *within* clusters. Maximizing this term alone is not sufficient as this would lead to a trivial partition with a single cluster. The role of the second term is rule out this trivial partition and others, with very few clusters. Specifically, the second term is, up to a multiplicative constant close to 1 and a small additive constant, the expected proportion of edges within clusters in the same graph but after having cut and shuffled all edges (i.e., in the associate configuration model). In this random graph (possibly a multi-graph), the expected number of edges between nodes $i, j$ is:

$$\begin{cases} \dfrac{d_i d_j}{2m-1} & \text{if } i \neq j, \\ \dfrac{d_i(d_i-1)}{2m-1} & \text{if } i = j. \end{cases}$$

Thus the expected proportion of edges within clusters in this random graph is:

$$\frac{1}{2m(2m-1)}\sum_{i,j\in V} d_i d_j \delta_{C(i),C(j)} - \frac{1}{2m(2m-1)}\sum_{i\in V} d_i = \frac{1}{2m(2m-1)}\sum_{i,j\in V} d_i d_j \delta_{C(i),C(j)} - \frac{1}{2m-1}.$$

To summarize, the modularity can be interpreted as the difference between proportion of edges within clusters in the real graph and in some random graph with the same degrees. This means that, if there is some clustering $C$ with high modularity, you don't observe many edges within clusters by chance: this must be due to the structure of the graph.

# 3 Sampling distribution

Consider the following sampling process: select an edge uniformly at random, then one of the two possible orderings of this pair with probability 1/2. Each node pair $i, j$ (in this order) is then chosen with probability:

$$p(i,j) = \frac{A_{ij}}{2m}.$$

This is a symmetric joint distribution with marginal distribution:

$$p(i) = \sum_{j\in V} p(i,j) = \frac{d_i}{2m}.$$

The modularity of any clustering $C$ can then be written as:

$$Q(C) = \sum_{i,j\in V} (p(i,j) - p(i)p(j))\delta_{C(i),C(j)}.$$

This is the difference between the probability of sampling an edge within a cluster and the probability of sampling a node pair (independently, under the marginal distribution) within a cluster.

Now, given some clustering $C$, this sampling process induces a joint distribution on the clusters:

$$p_C(k,l) = \sum_{i,j:C(i)=k,C(j)=l} p(i,j),$$

with marginal distribution:

$$p_C(k) = \sum_{l=1}^{K} p_C(k,l) = \sum_{i:C(i)=k} p(i).$$

We get:

$$Q(C) = \sum_{k,l=1}^{K} (p_C(k,l) - p_C(k)p_C(l))\delta_{k,l} = \sum_{k=1}^{K} (p_C(k,k) - p_C(k)^2).$$

Denoting by $m_k$ the number of edges in cluster $k$ and by $w_k$ the total degree in cluster $k$, we refer to as the *weight* of cluster $k$, we get:

$$Q(C) = \sum_{k=1}^{K} \frac{m_k}{m} - \sum_{k=1}^{K} \left(\frac{w_k}{w}\right)^2, \tag{1}$$

where $w = 2m$. The first term is still the proportion of edges within clusters. The second term is the Simpson index[1] associated with the probability distribution $w_1/w, \ldots, w_K/w$ (induced by uniform edge sampling), a classical measure of diversity in biology. The most diverse distribution is uniform over $\{1, \ldots, K\}$, leading to the minimum Simpson index $1/K$. We conclude that the modularity of any clustering with $K$ clusters cannot exceed $1 - 1/K$.

[1]Interpreting $p_C(k) = w_k/w$ as the proportion of individuals of species $k$, the Simpson index is the probability of getting two individuals of the same species when sampled uniformy at random in the total population [3].

# 4 Resolution

In view of (1), modularity is a quality metric balancing efficiency (first term) and diversity (second term). The first term tends to reduce the number of clusters (to improve efficiency) while the second tends to increase the number of clusters (to improve diversity). Maximizing modularity yields a clustering with some granularity that is hard to predict and impossible to control. In particular, the number of clusters cannot be too large as the second term is equal to $1/K$ for $K$ clusters of same weights, that vanishes for large $K$.

The resolution is a parameter $\gamma$ introduced for controlling the respective weights of efficiency and diversity in the modularity. We obtain the modularity at resolution $\gamma$:

$$Q_\gamma(C) = \frac{1}{2m} \sum_{i,j \in V} \left( A_{ij} - \gamma \frac{d_i d_j}{2m} \right) \delta_{C(i),C(j)}.$$

When $\gamma \to 0$, the first term dominates and the optimal clustering has only one cluter; when $\gamma \to +\infty$, the second term dominates and the optimal clustering has $n$ clusters (one per node). The standard modularity corresponds to the case $\gamma = 1$. Observe that $Q_\gamma(C) \in [-\gamma, 1 - \gamma/K]$ for a clustering $C$ with $K$ clusters. In particular, the best clustering is expected to contain $K > \gamma$ clusters. Setting the resolution parameter $\gamma$ (e.g., for some target number of clusters $K$) is a difficult problem in practice.

# 5 Weighted graphs

The modularity easily extends to weighted graphs. Let $A$ be the weighted adjacency matrix. Define the weight of node $i$ by:

$$w_i = \sum_{j \in V} A_{ij}.$$

The total weight of nodes is:

$$w = \sum_{i \in V} w_i = \sum_{i,j \in V} A_{ij}.$$

Observe that this is twice the total weight of edges as each edge is counted twice in this sum.

We define the modularity of some clustering $C$ as:

$$Q(C) = \frac{1}{w} \sum_{i,j \in V} \left( A_{ij} - \frac{w_i w_j}{w} \right) \delta_{C(i),C(j)}.$$

This definition extends that of unweighted graphs. In particular, we have:

$$Q(C) = \sum_{i,j \in V} (p(i,j) - p(i)p(j)) \delta_{C(i),C(j)},$$

where $p(i,j)$ and $p(i)$ are the edge and node sampling distributions induced by the weights. As above, this can be written in terms of cluster sampling distributions,

$$Q(C) = \sum_{k=1}^{K} (p_C(k,k) - p_C(k)^2).$$

We have the analogue of (1),

$$Q(C) = \sum_{k=1}^{K} \frac{w_k^{(E)}}{w^{(E)}} - \sum_{k=1}^{K} \left( \frac{w_k^{(V)}}{w^{(V)}} \right)^2, \tag{2}$$

where $w_k^{(E)}$ (resp., $w_k^{(V)}$) is the total weight of edges (resp., nodes) within cluster $k$ and $w^{(E)}$ (resp., $w^{(V)}$) is the total weight of edges (resp., nodes) in the graph. Note that $w = w^{(V)} = 2w^{(E)}$. The resolution parameter $\gamma$ can be included as in undirected graphs to control the granularity of the clustering.

# 6 Aggregation and self-loops

In view of (2), the modularity only depends on the weights of each cluster $k$, $w_k^{(E)}$ (total weight of edges) and $w_k$ (total weight of nodes). In particular, if two nodes $i, j$ belong to the same cluster, say cluster $k$, the modularity should remain the same by merging these two nodes, provided a self-loop is added to this new node.

Specifically, consider the aggregate graph where nodes $i$ and $j$ are replaced by a single node having a self-loop with weight $A_{ij}$ and an edge of weight $A_{ij'} + A_{jj'}$ with any node $j' \neq i, j$. Then the total weight of edges remains the same within each cluster; this is also the case of the total weight of nodes provided the weight of a self-loop is counted twice in the weight of a node (for unweighted graph, this amounts to consider that a self-loop contributes to 2 in the node degree).

This leads to the following definition of modularity for a graph with self-loops. Define the weight of node $i$ by:

$$w_i = 2A_{ii} + \sum_{j \neq i} A_{ij}.$$

The total weight of nodes is:

$$w = \sum_{i \in V} w_i = 2 \sum_{i \in V} A_{ii} + \sum_{i \neq j} A_{ij},$$

Observe that this is twice the total weight of edges. The modularity is then defined by:

$$Q(C) = \frac{1}{w} \sum_{i,j \in V} \left( A_{ij} - \frac{w_i w_j}{w} \right) \delta_{C(i),C(j)} + \frac{1}{w} \sum_{i \in V} A_{ii}.$$

The equality (2) remains valid, the self-loops being counted once in edge-based weights $w_k^{(E)}, w^{(E)}$ and twice in node-based weights $w_k^{(V)}, w^{(V)}$.

# 7 The Louvain algorithm

A classical approach to graph clustering consists in maximizing modularity, that is, in solving the problem

$$\max_C Q_\gamma(C),$$

where $\gamma$ is the resolution parameter. Although this optimization problem is NP-hard (even if $K$ is given, and in fact even in the simplest case $K = 2$), it is possible in practice to find good approximations of the optimal solution.

The most popular algorithm, known as the Louvain algorithm in name of the university of its inventors [1], is based on the following steps:

1. (Initialization) $C \leftarrow$ identity (each node has its own cluster).
2. (Maximization) While modularity $Q_\gamma(C)$ increases, update $C$ by moving one node from one cluster to another.
3. (Aggregation) If the number of clusters is less than the number of nodes, for each cluster, merge all nodes belonging to that cluster into a single node, update the weights accordingly and apply step 2 to the aggregate graph.

Observe that the algorithm ends in finite time as modularity increases strictly and there is a finite number of clusterings.

The outcome depends on the order in which nodes are considered at step 2; typically, nodes are considered in a cyclic way and the target cluster of each node is that maximizing the modularity increase. Step 3 forces the algorithm to explore more solutions by merging clusters, when modularity can no longer be increased by any local change of the clustering (one node moving from one cluster to another). The complexity of the algorithm depends mainly on the first maximization step (before the first aggregation), as all edges must be considered several times. The algorithm can be sped up by imposing a minimum modularity increase $\epsilon$ after one iteration over all nodes at step 2 before moving to step 3.

Let

$$w_{ik}^{(E)} = \sum_{j \neq i : C(j) = k} A_{ij}$$

be the total weight of edges incident to node $i$ and to nodes in cluster $k$ (different from node $i$). Then the variation in modularity induced by moving node $i$ from cluster $k$ to cluster $l \neq k$ is:

$$\begin{aligned}\Delta Q_\gamma &= \frac{1}{w^{(E)}}(w_{il}^{(E)} - w_{ik}^{(E)}) - \frac{\gamma}{{w^{(V)}}^2}\left((w_k^{(V)} - w_i)^2 + (w_l^{(V)} + w_i)^2 - {w_k^{(V)}}^2 - {w_l^{(V)}}^2\right), \\ &= \frac{1}{w}(w_{il}^{(E)} - w_{ik}^{(E)}) - \frac{\gamma w_i}{2w^2}(w_l^{(V)} - w_k^{(V)} + w_i).\end{aligned}$$

Let $l$ be the cluster maximizing this variation in modularity. If $\Delta Q_\gamma > 0$, then node $i$ must be moved from cluster $k$ to cluster $l$ and the variables are updated as follows:

$$w_k^{(V)} \leftarrow w_k^{(V)} - w_i, \quad w_l^{(V)} \leftarrow w_l^{(V)} + w_i,$$

and

$$\forall j \neq i, \quad w_{jk}^{(E)} \leftarrow w_{jk}^{(E)} - A_{ij}, \quad w_{jl}^{(E)} \leftarrow w_{jl}^{(E)} + A_{ij}.$$

Storing these node-cluster weights requires $O(m)$ memory. Checking whether each node must change its cluster and updating the corresponding variables requires $O(m)$ operations. The number of iterations depends on the graph.

# 8 Cluster ranking

Given some clustering $C : V \to \{1, \ldots, K\}$, it is worth assessing the quality of each cluster. We refer to the strength of cluster $k$ as the quantity:

$$\sigma_k = \frac{2w_k^{(E)}}{w_k^{(V)}}.$$

This can be interpreted as the proportion of inside weight in the total weight of the cluster. In particular, we have $\sigma_k \leq 1$, with equality if and only if cluster $k$ is disconnected from the rest of the graph.

Another interpretation is through random walks. Consider a random walk where the move from any node $i$ is selected at random among the neighbors of $i$ in proportion to the weights of the corresponding edges. This defines an irreducible Markov chain. The relative frequency of moves from node $i$ to node $j$ is exactly $p(i,j)$, while the relative frequency of visits to node $i$ (i.e., the stationary distribution), is $p(i)$. Similarly, the relative frequency of moves from cluster $k$ to cluster $l$ is $p_C(k,l)$, while the relative frequency of visits to cluster $k$ is $p_C(k)$. Observing that

$$p_C(k,k) = \frac{w_k^{(E)}}{w^{(E)}}, \quad p_C(k) = \frac{w_k^{(V)}}{w^{(V)}},$$

we get:

$$\sigma_k = \frac{p_C(k,k)}{p_C(k)} = p_C(k|k).$$

Thus $\sigma_k$ is the probability that, starting from cluster $k$, the random walk stays in cluster $k$ after one move. We expect this probability to be higher than $\pi_k \equiv p_C(k)$, the probability that the random walk lies in cluster $k$, because the random walk is already in cluster $k$. The modularity is exactly the weighted mean of the differences $\sigma_k - \pi_k$,

$$Q(C) = \sum_{k=1}^{K} \pi_k(\sigma_k - \pi_k).$$

In large graphs, $\sigma_k$ is typically much higher than $\pi_k$ since it is much more likely for the random walk to be in cluster $k$ if it was already in cluster $k$ in the previous state. It is then useful to introduce the resolution parameter $\gamma$ to make both quantities comparable, yielding

$$Q_\gamma(C) = \sum_{k=1}^{K} \pi_k(\sigma_k - \gamma\pi_k).$$

# References


[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[2] S. Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[3] E. H. Simpson. Measurement of diversity. *Nature*, 163(4148):688, 1949.