Graph mining
SD212
5. Clustering

Thomas Bonald

2018 – 2019

These lecture notes introduce some metrics and algorithms for graph clustering, a key technique in graph analysis, also known as community detection in the context of social networks. 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 $V = \{1, \dots, n\}$. We first 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 nodes connected through many short paths) tend to be in the same cluster. Formally, a clustering of the graph into K clusters is a 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) and one looks for the best clustering C irrespective of the 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:

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

where δ is the Kronecker symbol. Observe that $Q(C) \in [-1, 1]$ since

$$\sum_{i,j \in V} A_{ij} = \sum_{i \in V} d_i = 2m.$$

The higher the modularity $Q(C)$, the better the clustering C.

The modularity is the difference between two terms. The first term, equal to

$$\frac{1}{2m} \sum_{i,j \in V} A_{ij} \delta_{C(i),C(j)},$$

corresponds to the proportion of edges *within* clusters, which we want to be close to 1. Maximizing this term alone is not sufficient however as this would lead to a trivial partition with a single cluster. It is the role of

the second term to rule out this trivial partition as well as other partitions with a few dominant clusters. This second term, equal to

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

can be seen as the expected proportion of edges within clusters in the same graph but with edges cut and shuffled at random (i.e., in the associate configuration model). In this random graph (possibly a multi-graph), the expected number of edges between nodes i and j is approximately[1] equal to:

$$\frac{d_i d_j}{2m}.$$

Thus the modularity can be interpreted as the difference between the proportions of edges within clusters in the real graph and in some random graph with the same degrees, often referred to as the null model. This means that, if there is some clustering C with high modularity, the high proportion of edges within clusters is not due to chance: it must be due to the structure of the graph.

3 Sampling distribution

Another interesting interpretation of modularity is through node pair sampling. Consider the sampling of node pairs through uniform edge sampling. 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 two nodes independently (under the marginal distribution) within a cluster. If the clustering C is meaningful, you expect the former (that depends on the graph structure) to be much larger than the latter.

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

$$\forall k, l \in 1, \ldots, K, \quad p_C(k,l) = \sum_{i,j:C(i)=k,C(j)=l} p(i,j),$$

with marginal distribution:

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

We get:

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

[1] The exact value is $\frac{d_i d_j}{2m-1}$ for $i \neq j$ and $\frac{d_i(d_i-1)}{2m-1}$ for $i = j$.

Now let m_k the number of edges in cluster k and $v_k = \sum_{i:C(i)=k} d_i$ be the total degree in cluster k, which we refer to as the *volume* of the cluster. We have:

$$p_C(k,k) = \sum_{i,j:C(i)=k,C(j)=l} p(i,j) = \frac{m_k}{m}, \quad p_C(k) = \sum_{i:C(i)=k} p(i) = \frac{v_k}{2m},$$

so that

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

where $v = 2m$ is the volume of the graph (sum of node degrees). The first term appears explicitly as the proportion of edges within clusters. The second term is the Simpson index[2] associated with the probability distribution p_C, a classical measure of diversity / randomness in biology. The most diverse distribution is uniform over $\{1, \ldots, K\}$, leading to the minimum Simpson index $1/K$; the less diverse distribution is concentrated on a single value, corresponding to the maximum Simpson index 1. We conclude that the modularity of any clustering with K clusters cannot exceed $1 - 1/K$.

4 Resolution

In view of (1), modularity is a quality metric balancing *fit* (first term) and *diversity* (second term). The first term tends to reduce the number of clusters (to improve fit) while the second tends to increase it (to improve diversity). Maximizing modularity achieves a trade-off with some number of clusters K that is hard to predict beforehand. Note however that K cannot be too large as the second term (diversity) is equal to $1/K$ for K clusters with the same weight: this term vanishes for large K. This is known as the *resolution limit* of modularity.

To be able to control the number of clusters, especially to find partitions with a large number of clusters, the modularity can be modified as follows:

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

where γ is known as the resolution parameter. When $\gamma \to 0$, the first term (fit) dominates and the optimal clustering has only one cluter; when $\gamma \to +\infty$, the second term (diversity) 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 γ (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:

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

We refer to the volume of the graph as:

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

[2]Interpreting $p_C(k) = v_k/v$ as the proportion of individuals of species k, the Simpson index is the probability of getting two individuals of the same species when sampled uniformly at random from the total population [3].

where w is the total weight of edges (equal to m for unit weights).

We define the modularity of some clustering C as:

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

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),

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

where w_k is the *weight* of cluster k (total weight of edges within the cluster). The resolution parameter γ can be added as for undirected graphs to control the granularity of the clustering.

6 Aggregation and self-loops

In view of (2), the modularity only depends on the weight w_k (total weight of edges) and the volume v_k (total weight of nodes) of each cluster k. In particular, if two nodes belong to the same cluster, the modularity remains the same if these two nodes are *merged* into a single super-node, provided the cluster weight and the cluster volume are preserved.

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 $A_{ih} + A_{jh}$ with any other node $h \neq i, j$. Then the weight and volume of each cluster are preserved 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 volume of the graph is:

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

with w the total weight of edges:

$$w = \sum_{i \in V} A_{ii} + \sum_{i < j} A_{ij}.$$

The modularity is then defined by:

$$Q(C) = \frac{1}{2w} \sum_{i,j \in V} \left(A_{ij} - \frac{w_i w_j}{2w} \right) \delta_{C(i),C(j)} + \frac{1}{2w} \sum_{i \in V} A_{ii}. \tag{3}$$

The equality (2) remains valid, the self-loops being counted once in the cluster weights w_k and twice in the cluster volumes v_k. Observe that the clustering C maximizing the modularity does not depend on the self-loops, as each node is always in the same cluster as itself. Expression (3) is useful only for computing the exact value of modularity, especially after node aggregation due to the presence of self-loops.

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 γ 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 is in 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 C has changed, merge all nodes belonging to the same cluster into a single node, update the weights accordingly and go back to step 2; otherwise, stop.

Observe that the algorithm ends in finite time as modularity increases strictly at each step 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[3] after one iteration over all nodes at step 2 before moving to step 3.

Let

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

be the total weight of edges between node i and nodes in cluster k (different from node i). In view of (2), the variation in modularity induced by moving node i from cluster k to cluster $l \neq k$ is:

$$\begin{aligned}\Delta Q_\gamma &= \frac{1}{w}(C_{il} - C_{ik}) - \frac{\gamma}{v^2}\left((v_k - w_i)^2 + (v_l + w_i)^2 - v_k^2 - v_l^2\right), \\ &= \frac{1}{w}(C_{il} - C_{ik}) - 2\frac{\gamma w_i}{v^2}(v_l - v_k + w_i).\end{aligned}$$

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:

$$v_k \leftarrow v_k - w_i, \quad v_l \leftarrow v_l + w_i,$$

and

$$\forall j \neq i, \quad C_{jk} \leftarrow C_{jk} - A_{ij}, \quad C_{jl} \leftarrow C_{jl} + A_{ij}.$$

Observe that C_{ik} and C_{il} remain unchanged. Storing the 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.

[3] Such a threshold can also be used to identify so-called flipping nodes (nodes that can change cluster without any significant impact on modularity). The strength of their attachment to each target cluster can then be estimated through the total weight of edges to this cluster. By normalization, we get a probability distribution over the clusters, corresponding to a form of soft clustering.

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:

$$\rho_k = \frac{2w_k}{v_k}.$$

This can be interpreted as the proportion of the volume inside the cluster. In particular, we have $\rho_k \leq 1$, 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 provided the graph is connected. 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

$$p_C(k, k) = \frac{w_k}{w}, \quad p_C(k) = \frac{v_k}{v},$$

we get:

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

Thus ρ_k is the probability that, given that the random walk is in cluster k, it 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 $\rho_k - \pi_k$,

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

In large graphs, ρ_k is typically much higher than π_k unless the clustering C is coarse 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. The modularity at resolution γ, enabling clusterings of high resolution, becomes:

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

9 Directed graphs

There are several possible extensions to directed graphs. The simplest appraoch consists in viewing a directed graph of n nodes as a bipartite graph of $2n$ nodes, with biadjacency matrix A. This yields two clusterings, one per part of the bipartite graph, corresponding to nodes of the directed graph viewed as sources and destinations, respectively. The interpretations in terms of sampling and random walk are similar, but for the forward-backward random walk where moves occur alternately in forward and backward directions in the directed graph (corresponding to a regular random walk in the bipartite graph).

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.

[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.