Graph Mining
SD212
4. Graph Clustering

Thomas Bonald
Institut Polytechnique de Paris

2019 – 2020

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 [3] 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 denote by A the adjacency matrix and by $d = A1$ the vector of degrees.

We are interested in partitioning the set of nodes V into subsets called clusters so that "close" nodes (typically neighbors, having many neighbors in common) 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 the objective is to find 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:

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

where δ is the Kronecker symbol and v is the *volume* of the graph[1], defined by:

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

Observe that $Q(C) \in [-1, 1]$. The higher the modularity $Q(C)$, the better the clustering C.

The modularity is a quality metric balancing *fit* and *diversity* of the clustering. The first term of modularity, given by

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

is the proportion of edges *within* clusters[2]. This is a measure of the fit of the clustering to the graph. Maximizing this term alone is not sufficient however as this would lead to a trivial partition with a single

[1] Note that $v = 2m$ in the absence of self-loops.

[2] In the presence of self-loops, regular edges are counted twice while self-loops are counted once.

cluster. It is the role of the diversity term to rule out this trivial partition as well as other partitions with a few dominant clusters. The second term of modularity, equal to

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

can be seen as the proportion of edge weights within clusters in a weighted graph with adjacency matrix:

$$\hat{A}=\frac{dd^T}{v}.$$

Observe that the node degrees are preserved in the sense that $\hat{A}1 = d$. Thus the modularity can be interpreted as the difference between the proportions of edges within clusters in the real graph and in some reference graph with the same degrees, without any structure (nodes are connected in proportion to their degrees). 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

Another interpretation of modularity relies on the following experiment: sample two nodes at random and observe whether they are in the same cluster. Under edge sampling, each positive entry of the adjacency matrix is selected uniformly at random so that each node pair i, j (in this order) is sampled with probability:

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

This is a symmetric distribution with marginal distribution:

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

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

$$Q(C)=\sum_{i,j\in V}(p(i,j)-p(i)p(j))\delta_{C(i),C(j)}. \tag{1}$$

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.

Modularity can in fact be expressed directly through cluster sampling. Specifically, the probability of sampling clusters k and l (in this order) is:

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

This is a symmetric distribution with marginal distribution:

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

We get:

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

A more explicit form of modularity can be derived in the absence of self-loops. 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:

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

so that

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

The first term appears explicitly as the proportion of edges within clusters. The second term is the Simpson index[3] associated with the probability distribution p_C, a standard measure of diversity 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 Random walk

Consider a random walk in the graph. If the graph is connected, then edges are sampled uniformly at random in stationary regime, meaning that the probability that the random walker is in node i and moves to node j is $p(i,j)$. Moreover, the probability that the random walker is in node i is $p(i)$. In view of (1), the modularity is the difference between the probability that a random walk *stays* is the same cluster after a move and the probability that two independent random walks are in the same cluster.

5 Weighted graphs

The modularity easily extends to weighted graphs. The definition is the same, with A the weighted adjacency matrix and $d = A1$ the vector of node weights, and the interpretations in terms of sampling and random walk remain valid. In the absence of self-loops, we have the analogue of (2),

$$Q(C) = \sum_{k=1}^{K} \frac{w_k}{w} - \sum_{k=1}^{K} \left(\frac{v_k}{v}\right)^2, \tag{3}$$

where w_k is the weight of cluster k (total weight of edges within the cluster) and w the weight of the graph (total weight of edges).

6 Aggregate graph

It is interesting in practice to visualize the clustering through the corresponding aggregate graph, where each cluster is replaced by a single node. This is a weighted graph of K nodes with adjacency matrix:

$$A_C = MAM^T,$$

where M is the membership matrix, a binary matrix of dimension $n \times K$ with $M_{ik} = 1$ if and only if node i belongs to cluster k. Note that the weight of the edge between nodes k and l in the aggregate graph is the total weight of edges beween clusters k and l in the original graph; the aggregate graph has self-loops, with a weight of the self-loop of node k equal to the total weight of self-loops in cluster k in the original graph, plus *twice* the total weight of regular edges within cluster k in the original graph.

[3]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 [4].

Interestingly, the modularity is preserved by aggregation. Specifically, the modularity of the clustering C in the original graph is that of the trivial clustering (one cluster per node) in the aggregate graph. This follows simply on observing that p_C is the distribution induced by edge sampling in the aggregate graph.

7 The Louvain algorithm

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

$$\max_C Q(C),$$

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 $Q(C)$ increases, update C by changing the cluster of each node.
3. (Aggregation) If C has changed in step 2, merge all nodes belonging to the same cluster into a single node, update the weights 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 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, so excluding a possible self-loop at node i). The variation in modularity induced by moving node i from cluster k to cluster $l \neq k$ is:

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

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

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

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.

8 Resolution

Maximizing modularity achieves a trade-off between fit and diversity with some number of clusters K that is hard to predict beforehand. Note however that K cannot be too large as the second term of modularity (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:

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

where γ is known as the resolution parameter. When $\gamma \to 0$, the fit term dominates and the optimal clustering has only one cluter; when $\gamma \to +\infty$, the diversity 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 γ (e.g., for some target number of clusters K) is a difficult problem in practice.

9 Cluster strengths

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

$$\rho_k = \frac{1}{v_k} \sum_{i,j : C(i) = C(j) = k} A_{ij}.$$

In the absence of self-loops, we have:

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

The cluster strength can be interpreted as the proportion of the volume of cluster k inside the cluster. It is equal to the k-th diagonal element of the aggregate adjacency matrix A_C divided by the sum of row k. In particular, we have $\rho_k \leq 1$, with equality if and only if cluster k is disconnected from the rest of the graph. In the absence of self-loops, this is:

In terms of sampling and random walk, we have:

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

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. Observe that modularity is exactly the weighted average of the differences $\rho_k - \pi_k$,

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

10 Directed graphs

The notion of modularity can be extended to directed graphs [2]. Denoting by $d^+ = A1$ and $d^- = A^T 1$ the vectors of out-degrees and in-degrees, respectively, we get:

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

where v is the *volume* of the graph, defined by:

$$v = \sum_{i \in V} d_i^+ = \sum_{i \in V} d_i^- = \sum_{i,j \in V} A_{ij} = m.$$

The interpretation is the same, weights in the reference graph being proportional to the out-degree of the source and the in-degree of the destination.

Under edge sampling, each node pair i, j (in this order) is sampled with probability:

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

This distribution is no longer symmetric and has two marginal distributions:

$$p^+(i) = \sum_{j \in V} p(i,j) = \frac{d_i^+}{v} \quad \text{and} \quad p^-(i) = \sum_{j \in V} p(j,i) = \frac{d_i^-}{v}.$$

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

$$Q(C) = \sum_{i,j \in V} (p(i,j) - p^+(i)p^-(j))\delta_{C(i),C(j)} \tag{4}$$

or at cluster level:

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

Denoting by m_k the number of edges in cluster k and by $v_k^+ = \sum_{i:C(i)=k} d_i^+$ and $v_k^- = \sum_{i:C(i)=k} d_i^-$ the total out-degree and the total in-degree in cluster k, we get:

$$Q(C) = \sum_{k=1}^{K} \frac{m_k}{m} - \sum_{k=1}^{K} \frac{v_k^+ v_k^-}{v^2}. \tag{5}$$

The diversity term is now the dot product of the distributions of in-degrees and out-degrees in the clusters.

As for the Louvain algorithm, it can be adapted as follows. Let

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

be the total weight of edges between node i and nodes in cluster k (in any direction, but excluding a possible self-loop at node i). The variation in modularity induced by moving node i from cluster k to cluster $l \neq k$ becomes:

$$\begin{aligned}
\Delta Q &= \frac{1}{v}\left(C_{il} - C_{ik} - \frac{1}{v}\left((v_l^+ + d_i^+)(v_l^- + d_i^-) + (v_k^+ - d_i^+)(v_k^- - d_i^-) - v_l^+ v_l^- - v_k^+ v_k^-\right)\right), \\
&= \frac{1}{v}\left((C_{il} - C_{ik}) - \frac{d_i^+}{v}(v_l^- - v_k^- + d_i^-) - \frac{d_i^-}{v}(v_l^+ - v_k^+ + d_i^+)\right),
\end{aligned}$$

with updates:

$$v_k^+ \leftarrow v_k^+ - d_i^+,\ v_k^- \leftarrow v_k^- - d_i^-, \quad v_l^+ \leftarrow v_l^+ + d_i^+,\ v_l^- \leftarrow v_l^- + d_i^-$$

and

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

11 Bipartite graphs

A bipartite graph $G = (V_1, V_2, E)$ with biadjacency matrix B of dimension $n_1 \times n_2$, with $n_1 = |V_1|$, $n_2 = |V_2|$, can be seen either as an undirected graph with adjacency matrix:

$$A = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix}$$

or as a directed graph with adjacency matrix:

$$A = \begin{bmatrix} 0 & B \\ 0 & 0 \end{bmatrix}$$

The Louvain algorithm can then be applied to either graph, and the clustering derived from that of nodes in V_1 (the first n_1 nodes of the graph). The interest of using the directed graph is that the corresponding reference graph (associated with the second term of modularity) has only edges from the n_1 first nodes to the last n_2 nodes, corresponding to edges between V_1 and V_2 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] N. Dugué and A. Perez. Directed louvain: maximizing modularity in directed networks. Technical report, Université d'Orléans, 2015.

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

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