

1 Dense bipartite subgraphs

We previously looked at communities as simply dense subgraphs graphs, or as subgraphs such that each node has a large fraction of its edges inside. Another view, taken in [3], starts from the hubs and authorities model. It argues that a structure of densely linked hubs and authorities is a common feature of communities. Such a core, a *dense bipartite graph*, can be considered the “signature” of a community

Ideally, we would like to enumerate all such signatures, and expand them to communities. However, the complexity of doing so would be prohibitive. In fact, even finding just one large complete bipartite graph is NP-hard. However, when the given graph is dense enough, it always has a large complete bipartite subgraph.

Lemma 1 *If a bipartite graph has $\Omega(b^{1/3}n^{5/3})$ edges, it contains a $K_{3,b}$, i.e., a complete bipartite subgraph with 3 nodes on one side, and b nodes on the other.*

Proof. For each node v on the right side, we write $\delta(v)$ for the set of its neighbors, and $d(v) = |\delta(v)|$ for its degree. Each node v is labeled with each 3-element subset T of $\delta(v)$ (i.e., with all $T \subseteq \delta(v)$, $|T| = 3$). Notice that each node thus has many labels, namely $\binom{d(v)}{3}$. Hence, taken over all nodes on the right side, the total number of labels is $\sum_v \binom{d(v)}{3}$. We assumed in the statement of the lemma that $\sum_v d(v) = \Omega(b^{1/3}n^{5/3})$. The total number of labels is minimized when all $d(v)$ are equal, i.e., $d(v) = \Omega(b^{1/3}n^{2/3})$. Even then, the number of labels is

$$\sum_v \binom{d(v)}{3} = n \binom{\Omega(b^{1/3}n^{2/3})}{3} = n\Omega(bn^2) = \Omega(bn^3). \quad (1)$$

But the total number of distinct labels is only $\binom{n}{3} = O(n^3)$. Hence, by the Pigeonhole Principle, some label must appear at least b times. The b nodes on the right side sharing the label, and the three nodes on the left side who form the parts of the label, together form a $K_{3,b}$. ■

By considering a -tuples instead of triples for labels, we can obtain the generalization that any bipartite graph with $\Omega(b^{1/a}n^{2-1/a})$ edges contains a $K_{a,b}$.

While it is interesting to know that sufficiently dense graphs will contain a $K_{a,b}$, it does not necessarily help us in finding one, in particular if the graph is not dense. For large a and b , the problem is NP-hard in general, but we may still be interested in speeding up the search for smaller, and practically important, values, such as finding $K_{3,6}$ graphs. By brute force (trying all 9-tuples of nodes), this would take $\Theta(n^9)$ steps. A first and simple improvement is given by realizing that we only need to look at triples of nodes on one side. Given nodes v_1, v_2, v_3 , we can take the intersection of their neighborhoods $\bigcap_i \delta(v_i)$. If the intersection contains at least b elements, then a $K_{3,b}$ has

been found, else those three nodes cannot be part of a $K_{3,b}$. This reduces the running time to $\Theta(n^4)$.

The ideas underlying this improvement can be extended further. Obviously, any node of indegree less than 3 can be pruned, and similarly for outdegrees less than b . Once nodes have been pruned, we can iterate, as the degree of other nodes may have been reduced. In addition, if a node reaches indegree exactly 3 (or outdegree exactly b), it can be verified easily if it and all its neighbors form a $K_{3,b}$, after which they can either be reported (and pruned), or just pruned. These heuristics, while not useful in a worst-case scenario, help a lot in practice.

Building up large bipartite graphs can be likened to finding dense areas of a 0-1 matrix, which is a task known as *association rule mining* in the data mining community. A common approach there is to take simple rules, and combine them into larger rules. The idea is that any subgraph of a larger complete (or dense) graph must itself be complete (or dense). Hence, looking only at combinations of small dense graphs rules out a lot of unnecessary attempts. By starting from $K_{1,1}$ graphs, extending them to $K_{1,2}$ and $K_{2,1}$, then to $K_{2,2}$, $K_{3,1}$, and $K_{1,3}$, etc., we make sure to only look at relevant data, which leads to a lot of speedup in practice (though again no theoretical guarantees in the case of dense graphs).

2 Spectral techniques

Another approach to finding sub-communities is to look at higher eigenvectors of the adjacency matrix (or cocitation matrix), and nodes with large positive or negative entries in those eigenvectors (hub-authority weights) [2]. Such nodes intuitively identify hubs and authorities of sub-communities.

Some intuition why this may work well can be gained from the following matrix B , representing a graph.

$$B = \begin{bmatrix} 2 & 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 3 & 3 & 3 \end{bmatrix}$$

The largest eigenvalues of B are $\lambda_1 = 9$ and $\lambda_2 = 4$, and the corresponding eigenvectors are $\vec{e}_1 = [0 \ 0 \ 1 \ 1 \ 1]^T$ and $\vec{e}_2 = [1 \ 1 \ 0 \ 0 \ 0]^T$, respectively. The sub-communities found are $K_{3,3}$ and $K_{2,2}$. In general, eigenvectors tend to separate out different disconnected (or only sparsely connected) components. Thus, they can be used to discover some corresponding community structure.

3 Modularity

Modularity [1, 4] measures when a specific division into communities is a good one, i.e., when there are many edges within communities and few edges between them. That is, modularity indicates how much non-random characteristic a graph exhibits.

For a graph $G = (V, E)$, we consider a partition $\mathcal{P} = \{S_1, \dots, S_k\}$, and write $e(S_i) := \{(u, v) \in E \mid u, v \in S_i\}$ for the number of edges inside component S_i for $i = 1, 2, \dots, k$. Then, the number of

edges within communities is

$$e(\mathcal{P}) := \sum_{i=1}^k |e(S_i)|.$$

So, the fraction of edges inside communities is

$$\frac{e(\mathcal{P})}{m} = \frac{\sum_{i=1}^k |e(S_i)|}{m},$$

where m denotes the number of edges in G . In order to characterize if a partitioning is good, we would like to characterize how much of the actual structure it captures. We can do this by comparing the fraction of edges inside communities to the fraction of edges that would end up inside communities if the graph were random (conditional upon its degree distribution). If the partitioning is good, then we would expect the former to be significantly larger than the latter.

More formally, we write $d(S_i) := \sum_{v \in S_i} d(v)$ for the number of edge endpoints inside S_i . Then, the expected number of edges entirely in S_i is

$$\frac{1}{2}d(S_i)\frac{d(S_i)}{2m} = \frac{d(S_i)^2}{4m},$$

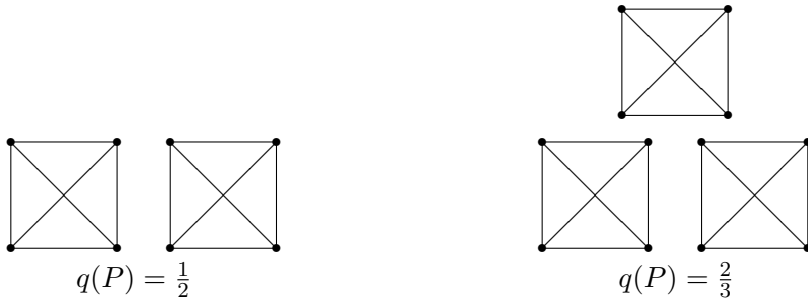
because each of the $d(S_i)$ edge endpoints in S_i has a probability of $d(S_i)/2m$ of having its random partner be in S_i , and each edge in S_i is counted twice (once per endpoint). We define the modularity $q(\mathcal{P})$ of a partition \mathcal{P} as the difference between the actual fraction of edges inside, and the expected fraction:

$$q(\mathcal{P}) = \frac{e(\mathcal{P})}{m} - \frac{\sum_{i=1}^k \frac{d(S_i)^2}{4m}}{m} = \frac{4me(\mathcal{P}) - \sum_{i=1}^k d(S_i)^2}{4m^2}$$

The goal of any algorithm is then to find a clustering of large modularity. Notice that one (desirable) feature here is that the number of clusters k need not be specified. It is part of the “right” solution.

In [1], it is found that when the modularity $q(\mathcal{P})$ is above about 0.3, the partition \mathcal{P} represents a significant community structure. Beyond some empirical results, not much is known about this (fairly new) measure. It is not even known if it is NP-hard to find a clustering maximizing modularity, though that seems likely.

As an example, observe the following two graphs. In these graphs, the clustering of maximum modularity is actually achieved by the “natural” clustering into the connected components. The corresponding modularity values are given below.



References

- [1] C. Moore A. Clauset, M. Newman. Finding community structure in very large networks. *Phys. Rev. E* 70, 066111, 2004.
- [2] P. Raghavan D. Gibson, J. Kleinberg. Inferring web communities from link topology. In *Proc. 9th ACM Conference on Hypertext and Hypermedia*, 1998.
- [3] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *8th International World Wide Web Conference*, 1999.
- [4] M. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* 69, 066133, 2004.