From Louvain to Leiden: guaranteeing well-connected communities

Traag et al. [2] proposes a new algorithm for community detection that improves connectedness. In particular, they show that compared to the Louvain algorithm, it has better accuracy. Moreover, they prove theoretical guarantes about the connectedness of the output.

First, they describe modularity, an objective function which is given by the formula

$$\mathscr{H} = \frac{1}{2m} \sum_{c} \left(m_c - \gamma \frac{K_c^2}{2m} \right)$$

where c denotes a community, m the edges and K the sum of degrees. Note that the expected number of edges in a community is $\frac{K_c^2}{2m}$, so the algorithm maximizes differences between the expected and actual. Higher resolutions (γ) lead to more communities, while lower resolutions lead to fewer.

Similarly, the Constant Potts Model (CPM) is given by

$$\mathscr{H} = \sum_{c} \left(m_c - \gamma \frac{n_c}{2} \right)$$

The density within and between communities are separated by the threshhold γ.

For both models, higher resolutions lead to more communities, while lower resolutions lead to fewer. TODO: explain why.

The Louvain algorithm [1] maximizes Modularity or CPM in by repeating two stages.

1. Local moving of nodes
2. Aggregation of the network

Stage I is a greedy step: move each vertex into the community s.t. maximizing the increase in score. Stage II then collapses each communities into a nodes. The algorithm runs until convergence (when Step I does nothing). At this point, merging two communities never increases the score, and neither does moving a node.

First, they show a pathological example of when Lovain produces a disconnected cluster. Suppose we have clusters c_1 and c_2, with vertex v a vertex cut in c_1. Then, because of the independent movement, v can be moved into c_2 and so disconnecting c_1. Moreover, the more connected c_1 is, the less likely it is for it to be split. TODO: find a better example than this

The Leiden algorithm introduces an additional step of *refining* the partition $\mathcal{P}$. $\mathcal{P}_{\text{refined}}$ is created in the following way:

0. $i = 0$, and P_i is the singleton partition
1. For each community in $\mathcal{P}$,
2. (a) For each node, consider all communities in $P_i\big|_{\mathcal{P}}$ s.t. the objective increase

   (b) Pick a community uniformly and merge
3. Increment i, go to 1.

References

[1] Vincent D Blondel et al. “Fast unfolding of communities in large networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 2008), P10008. DOI: 10.1088/1742-5468/2008/10/P10008. URL: https://dx.doi.org/10.1088/1742-5468/2008/10/P10008.

[2] V. A. Traag, L. Waltman, and N. J. van Eck. “From Louvain to Leiden: guaranteeing well-connected communities”. In: *Scientific Reports* 9.1 (Mar. 2019), p. 5233. ISSN: 2045-2322. DOI: 10.1038/s41598-019-41695-z. URL: https://doi.org/10.1038/s41598-019-41695-z.