Modularity based community detection in hypergraphs

Presented by: Ian Chen

University of Illinois Urbana-Champaign

August 15, 2026

Why Networks

- Modeling any discrete set of objects that are related to each other
- Traditional graphs represent dyadic relationships
  - Adjacency is a measure of structure; describe connections
  - Examples [1] friendship, interactions, information or material flow, . . .
- Hypergraphs can be more faithful representation of structure:
  - co-authorship: multiple authors on the same paper
  - friendship: multiple people within the same friend-group
- Often, study *two-section* of the graph: polyadic relationships become cliques

Graphs and hypergraphs

- $G = (V, E)$, where edges are sets of *multi-sets* of size 2
  - A loop consists of an edge $\{ v, v \}$
  - Degree, adjacency, incidence
- Graphs are 2-uniform, i.e. all edges of same cardinality (2)
- Hypergraphs relax the restriction away from all edges being same cardinality
  - Rank is the size of largest edge
  - $[H]_2$ is 2-section of hypergraph H
- Intra-community edges have (strict) majority of nodes within community
  - Homogeneity of e measures proportion of nodes in that community

h-ABCD

- ABCD was developed to be an improvement over LFR
- Creates power-law community sizes and degrees
- ξ is the mixing paramter, proportion of inter-community edges
- Structure of intra-community edges controlled by $w_{c,d}$
  - Proportion of size d intra-community edges that have c nodes inside community
  - $c \leq d$, by edge being of size d
  - $d/2 < c$, by being an intra-community edge

h-Modularity

- Modularity of a partition $\mathcal{A}$ measures *edge contribution* - *degree tax*
  - More intra-community edges reinforces the belief that $\mathcal{A}$ is good
  - Compare to null model (high volume sets have a baseline amount of edges)
- Modularity has *resolution* parameter, dictating belief of edge contribution
- In h-Modularity, need a parameter for each slice, of (c, d) edges

$$q_H^{c,d}(\mathcal{A}) = \frac{1}{|E|} \sum_{A_i \in \mathcal{A}} \left(e_H^{c,d}(A_i) - \mathbf{Ev}_{H'}\left(e_{H'}^{c,d}(A_i) \right) \right)$$

$$q_H(\mathcal{A}) = \sum_{c,d} \eta_{c,d} q_H^{c,d}(\mathcal{A})$$

- Recommend $\eta_{c,d} \sim (c/d)^\tau$, for user-specified τ

h-Louvain

- Louvain algorithm is iterative greedy procedure (similar to LPA):
  1. Move each node to a community to maximize modularity
  2. Contract the graph into supernodes and recurse
- h-Modularity has a large *cold-start* problem
  - Edge contribution is small until the community has enough nodes
- Solutions:
  - Change the algorithm to move multiple nodes at a time
  - Change the optimization to reduce the cold-start problem

$$q_H(\mathcal{A}, \alpha) = \alpha q_H(\mathcal{A}) + (1 - \alpha) q_{[H]_2}(\mathcal{A})$$

h-Louvain (cont.)

- Picking the right modularity function to optimize matters:
- Use prior beliefs:
  - Smaller τ means that we accommodate more noise, but less strong
  - Larger τ is stricter, and leads to smaller clusters
- Run Louvain on the 2-section as exploratory data analysis to fit τ

Discussions

- What kind of networks should we want to study?
  - Only unweighted simple graphs?
  - What about attributes? Temporal? Hypergraphs?
- Descriptive versus inferential?
  - Finding community structure that best describes the network?
  - Or, do we want to infer some underlying process that generates the network?
- Are the problems with modularity fixed if we pick the right resolution?
  - We know Leiden-CPM suffers from resolution limit already (heuristic).

References I

[1] Albert-Laszlo Barabasi. "Network science". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1987 (Mar. 2013), p. 20120375. ISSN: 1471-2962. DOI: 10.1098/rsta.2012.0375.

[2] Bogumił Kamiński, Paweł Prałat, et al. *Mining complex networks*. Chapman and Hall/CRC, 2021.

[3] Bogumil Kaminski et al. "Modularity based community detection in hypergraphs". In: *Journal of Complex Networks* 12.5 (2024). ISSN: 2051-1329. DOI: 10.1093/comnet/cnae041.