Community Detection in Heterogeneous and Evolving Networks

—— A Clustering Combined Heuristic Approach

Honglin Bao[1,2], Yiwen Hu[1], and Yumeng Wen[1]
1 Department of Computer Science and Engineering
2 NSF Beacon Center for the Study of Evolution in Action
Michigan State University
* Authors listed in Alphabetical order

Mainstream Approaches for Community Detection

- In the study of complex networks, a network is said to have community structure if the nodes of the network can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally.

  - Heuristic approaches: GA, multi-objective approach, etc.
  - Optimization: Spectral Clustering in our lecture; Modularity-based Optimization, etc.
  - Supervised Approaches: clustering, label propagation, etc.
  - Deep Learning: GNN, Graph Embedding, etc.

Drawbacks

- NP-Hard / NP-Complete
- Parallel
- Convergence Speed and Convergence Rate
- Evolvability and Heterogeneity

Our Approaches and Innovation

- NP-Hard / NP-Complete and Parallel

  Heuristic Approaches, e.g., GA.

- Convergence Speed and Convergence Rate

  The ability to find better offsprings; efficient crossover operator; stochastic generation of initialized networks; fully random mutations.

  Clustering-inspired crossover.

- Evolvability and Heterogeneity

  Define a new mechanism to generate dynamical, heterogeneous, and evolving structure.

Evolvability and Heterogeneity

- Cut a period of time ***T*** as multiple discrete time points.
- The evolving network structure can be modeled as a series of snapshots W_T =[W(1),W(2),…, W(T)]
- The goal is to find the evolving community series in ***T*** discrete time series, Z_T ={Z(1),Z(2),…,Z(T)}
- The community structure at ***T*** is related to:

1) the community structure at (T-1);

2) the network sequential relationship between T-1 and T.

$$L_{Z(t)} = \mathrm{lb}P(W^{(t)} \mid Z^{(t)}) + \mathrm{lb}P(Z^{(t)} \mid Z^{(t-1)})$$

$$Z^{*(t)} = \mathrm{argmax}_{Z(t)} P(W^{(t)} \mid Z^{(t-1)})$$
$$= \mathrm{argmax}_{Z(t)} P(W^{(t)} \mid Z^{(t)}) P(Z^{(t)} \mid Z^{(t-1)})$$

The relationship between community structure and real network at snapshot T — a static method

The relationship between two continue community structures at two sequential snapshots — Introduce Markov

Heterogenerity: Degree (hub node and ordinary node), direction, and edge weight.

Yang, T., Chi, Y., Zhu, S., Gong, Y., and Jin, R. (2009, April). A bayesian approach toward finding communities and their evolutions in dynamic social networks. In *Proceedings of the 2009 SIAM International Conference on Data Mining* (pp. 990-1001). Society for Industrial and Applied Mathematics.

Problem Define

- Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules. Modularity is often used in optimization methods for detecting community structure in networks.
- **It has been shown that modularity suffers a resolution limit and, therefore, it is unable to detect small communities.**

$$Q = \sum_i (e_{ii} - a_i^2)$$

- This quantity measures the fraction of the edges in the network that connect vertices of the same type ~i.e., within community edges minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices.

Overview of Algorithmic Framework

- 1. Markov Evolving and Heterogeneous Network Generation
- 2. Genetic Algorithm for Community Detection
- 2.1 Encoding
- 2.2 Clustering-based Crossover
- 2.3 Mutation
- 2.4 Fitness Evaluation (Q) and Selection

Markov Generation

- **Key idea: The agent's walk can be viewed as a stochastic process defined based on the links' attributes**
- Let p_{ij} be the probability of the agent walking from node i to its neighbor, node j.

$$p_{ij} = \frac{1}{d_i},$$

- Probability that the agent starting from node i can eventually arrive at a specific sink node t within l steps.

$$P_t^l(i) = \begin{cases} 1, & i = t \\ \sum_{\langle i,j \rangle} p_{ij} \cdot P_t^{l-1}(j), & i \neq t \end{cases} \quad (1 \leq i, t \leq n)$$

$$\forall_{i \in C_t} \forall_{j \notin C_t} \{P_t^l(i) > P_t^l(j)\}, \quad 1 \leq i, j, t \leq n$$

Yang, B., Cheung, W., and Liu, J. (2007). Community mining from signed social networks. *IEEE Transactions on Knowledge and Data Engineering*, *19*(10), 1333-1348.

Heterogeneity

- Edge Weight
- Directed Graph or Not.

$$p_{ij} = \frac{1}{d_i}, \longrightarrow p_{ij} = \frac{w_{ij}}{k_i}$$

- Wij: the weight of link i to j (Notice to figure out i to j VS. j to i)
- Ki: the total weighted out-degree of node i, given by

$$k_i = \sum_{i \to j} w_{ij}.$$

Recursive Generation Algorithm

Goal: Generate an evolving network with 1) approximate community structure (it is meaningless to detect random network); 2) degree, direction, and weight heterogeneity

- 1. Select a node t randomly
- 2. Based on t, calculate transition probability of all other nodes.
- 3. Order these probabilities from high to low, store them in an FIFO queue.
- 4. For all elements in this queue

    4.1 select TOP element in this queue

    4.2 calculate Q

    4.3 find whether exists a node which increases Q, if many, select the node with the largest Q incremental.

5. Yes → Split into two sub-networks. No → Find again until visiting all elements.

6. Recursion until splitting the network into multiple clusters.

Detection Process

- Encoding
- Fitness Define
- Clustering Inspired Crossover
- Selection and Mutation

Encoding and Fitness

- Fitness: Q
- Encoding:

Assume 10 nodes, 4 communities.

An individual is an encoding sequence. Genetic Operators work on individual.

(0,1,2,3,1,2,3,1,1,0) --- node 1,10 are within a community.

Drawback: 1. The number of community is a hyper-parameter.

2. Crossover: (1 1 1 2 2 2) X (2 2 2 1 1 1)=(1 1 1 1 1 1) + (2 2 2 2 2 2)

(1 1 1 2 2 2), (2 2 2 1 1 1) represent different individuals but actually indicate the same community structure.

"The building block hypothesis"

Clustering-inspired Crossover

Crossover on node → Crossover on edges

Key idea: Find cut

1. Select M individuals.
2. Rank all edges in high cut → low cut, store them in an FIFO queue.
3. For all elements in this queue

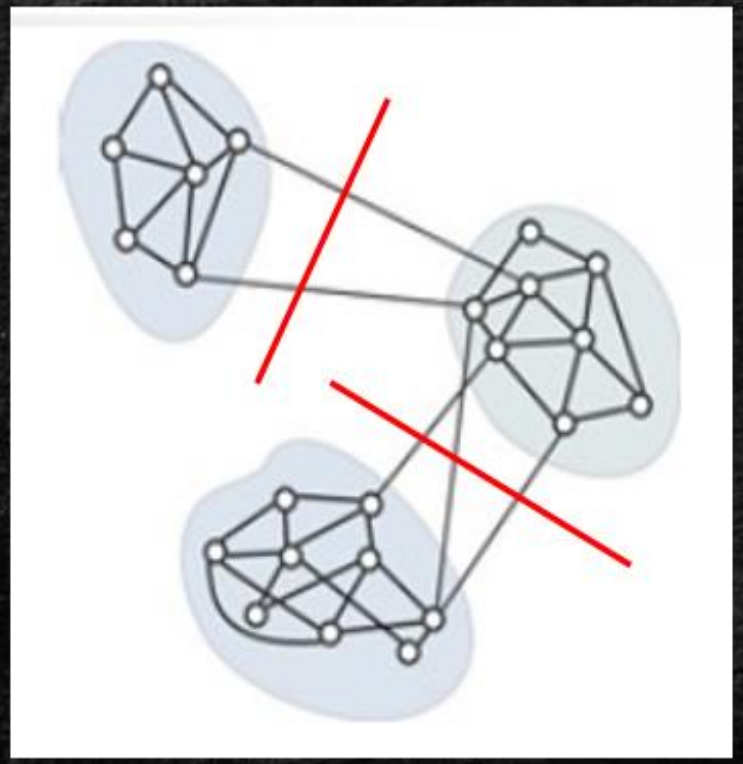
   3.1 Each node is initialized as a community.

   3.2 Select TOP element

   3.3 If this edge connects two different communities. Then combine them as a new community. If not, back to 3.2.

   3.4 Select the individual with the highest Q as new individual after crossover.

   3.5 Population = Original Population U New Individual. Back to 1.



Mutation

- Suppose 4 communities, 10 nodes.

(0,1,2,3,1,2,3,1,2,0)

Majority Voting Mutation

Suppose node 0 has three neighbors, node 1 (1), node 2 (2), and node 4 (1), respectively.

Original Node 0 (No.0 community) → New Node 0 (No. 1 community)

Weighted Voting Mutation

Selection: Evolutionary Strategy

- **Elitist ES:**

(1+1)-ES and (μ+λ)-ES

- **Non-Elitist ES:**

(μ, λ)-ES and (μ / μ_I, λ)-ES (multi-recombination)

Here we adopt (1+1)-ES: The simplest evolution strategy operates on a population of size two: the current point (parent) and the result of its mutation. Only if the mutant's fitness is at least as good as the parent one (Q), it becomes the parent of the next generation. Otherwise the mutant is disregarded.

Experiments

- Base on Computer Generated Network proposed by M. E. J. Newman, UMich *.
- Each graph consists of n = 128 vertices divided into four groups of 32. Each vertex has on average z_in edges connecting it to members of the same group and z_out edges to members of other groups, with z_in and z_out chosen such that the total expected degree z_in+z_out = 16.
- As z_out is increased from small values, the resulting graphs pose greater and greater challenges to the community-finding algorithm.

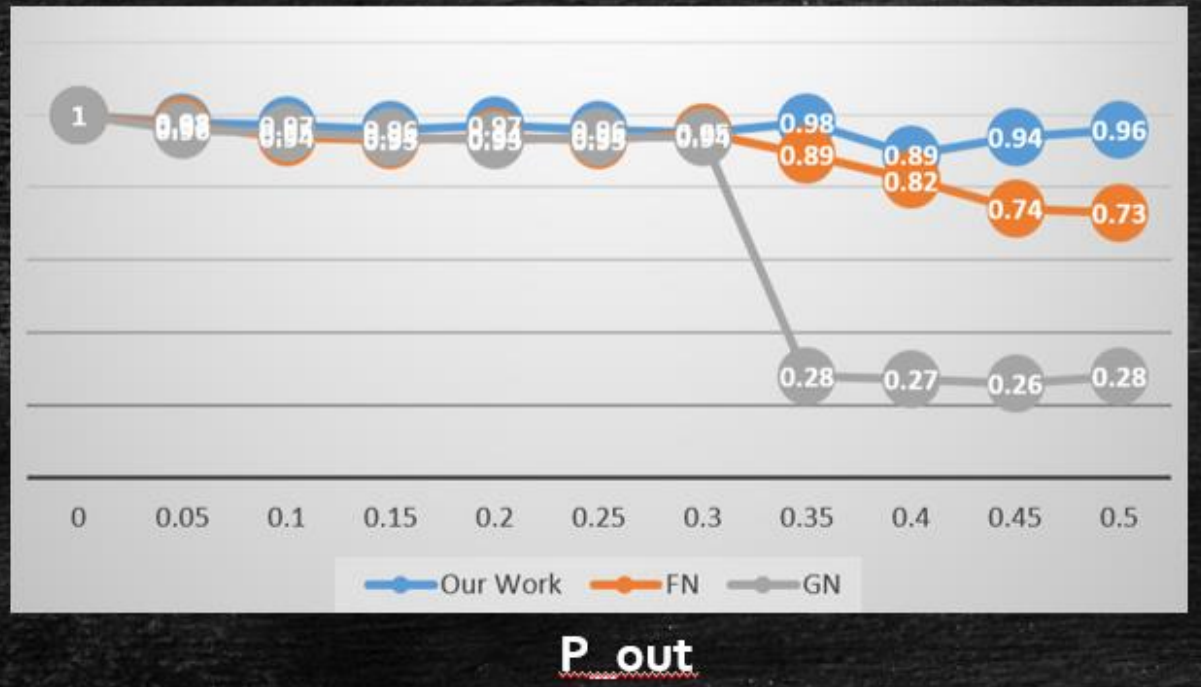
Newman, M. E. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, *69*(6), 066133.

Evaluation

- Calculate the fraction of vertices correctly classified and took it as a measure of clustering accuracy.
- Compare our results with the Girvan-Newman algorithm (GN) and Newman-fast algorithm (FN) who are quite classical and frequently referenced at present.

Newman, M. E. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, *69*(6), 066133.

Girvan, M., & Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, *99*(12), 7821-7826.



Future Work

- 1. Resolution limit → How to define new fitness?
- 2. Real World Evaluation?
- 3. How to measure node diversity of generated networks?
- 4. Time Complexity? Comparison?
- 5. More Experiments?
- 6. Parameter Selection?

In this work, population size: 100, time step: 500, crossover candidate M: 5, mutation rate: 10% ……

Can we adjust them to get better performance?

Code Available: https://github.com/hlbao/communitydetection

Q & A, thanks!