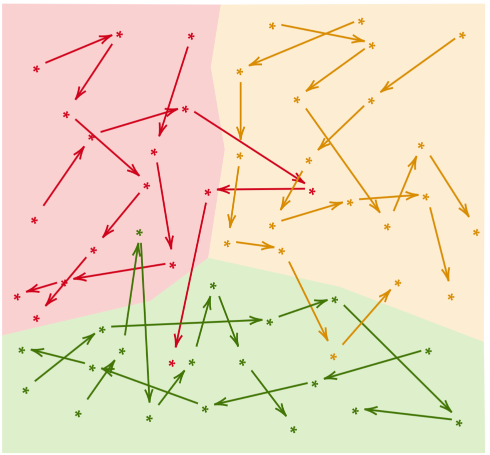
Community Detection in Trajectories: Girvan–Newman & Louvain

Yang Tan Jake Jasmer Tony Ni Aidan Roessler

Carleton College

Introduction

This research focuses on detecting **activity areas** of moving objects within a defined region by representing their **trajectories** as **graphs**, where **timestamped positions** serve as nodes. We compare two algorithms for **community detection** based on their effectiveness in identifying distinct areas.

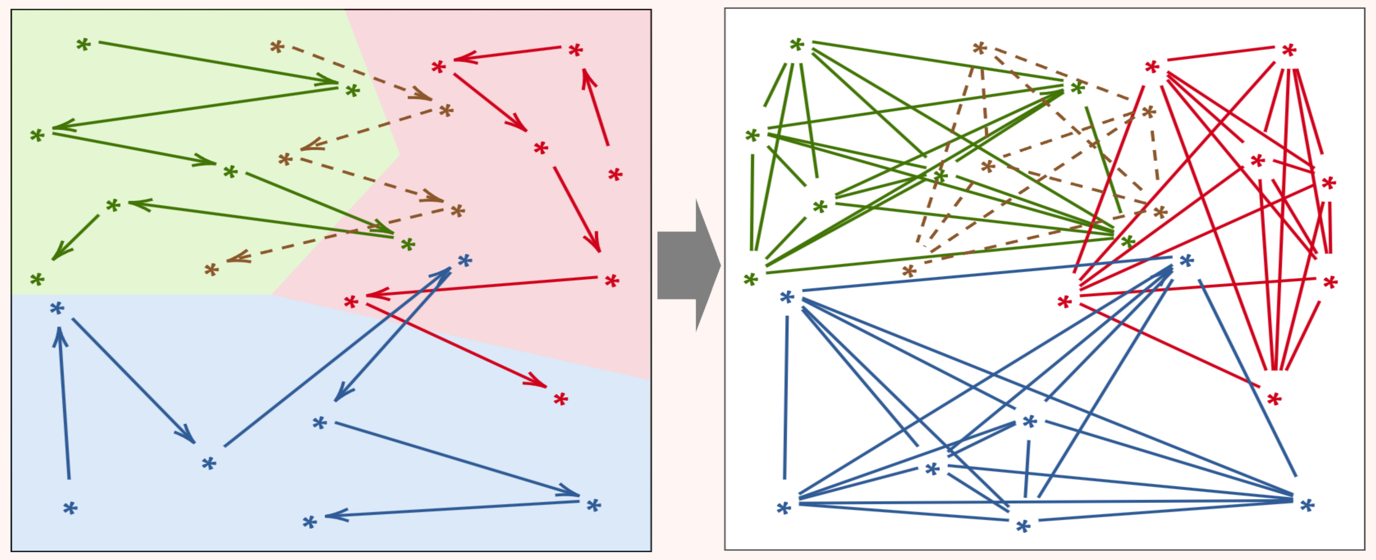


Community Detection Overview

- **Input:** Graph $G = (V, E)$ with nodes as timestamps and edges as paths.
- **Output:** Clusters of nodes with the highest internal connectivity.

Methodology: Trajectory Simulation

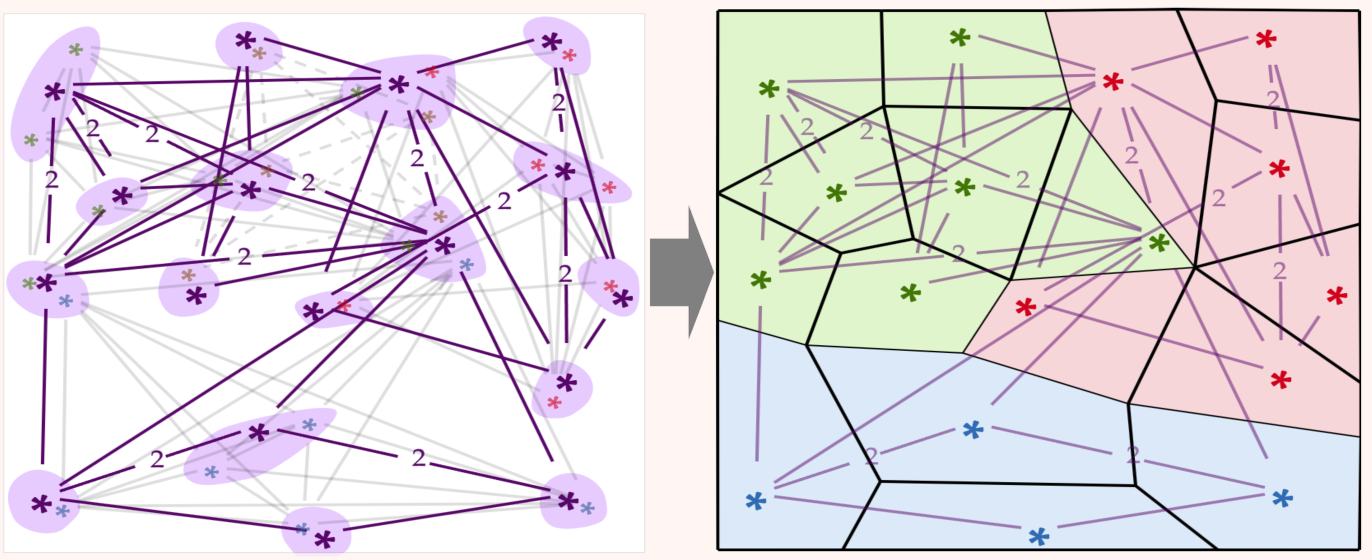
The simulation is replicated from paper[4].



We divide a rectangular region into n areas and simulate two types of moving objects to generate trajectories (each of length **10**):

- **Type 1** ($10n$ **objects**): Remains in **one** area **90%** of the time and moves randomly **10%** (red, green, blue in the figure).
- **Type 2** ($\frac{10n}{2}$ **objects**): Remains in **two** areas **90%** of the time and moves randomly **10%** (brown, dotted in the figure).

Each trajectory, consisting of **10 steps**, is then converted into an **undirected complete graph**, where every step is represented as a **node** containing **location information**.



To analyze activity areas, we:

- Apply k**-means clustering** (with $k = 0.4\times$ number of nodes) to merge nearby vertices, forming a connected weighted graph.
- Run a **community detection algorithms** to assign nodes to communities.
- Use **Voronoi polygons** to visualize the detected communities and compare them to the original regions.

Community Detection Algorithms

Expected weight between nodes i & j (probability of two half-edges being connected)

Actual weight between nodes i & j

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i,\, c_j)$$

Total number of half-edges

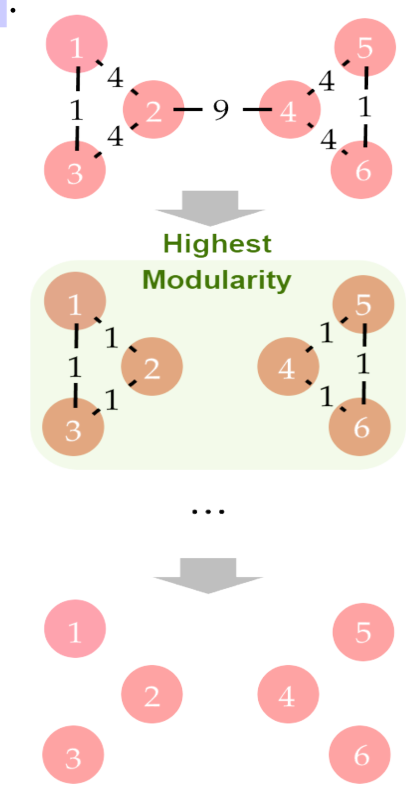
1 if nodes i & j are in the same community, 0 otherwise

The objective function for community detection is **modularity** Q, which we aim to **maximize**.

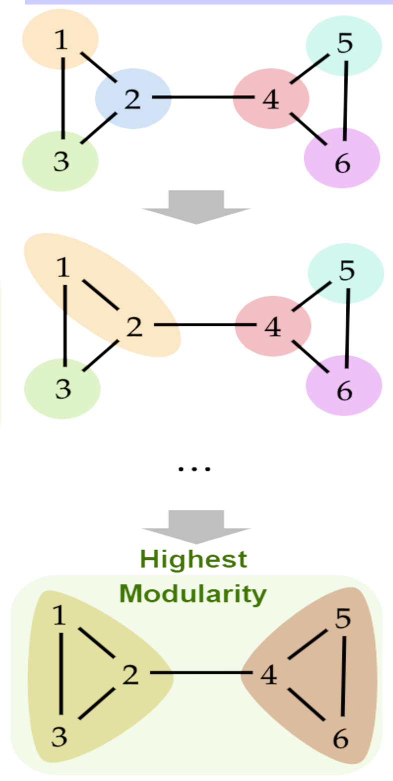
Modularity $Q =$ Actual connectivity within communities $-$ Expected connectivity.

Girvan-Newman Algorithm:

- Calculate edge betweenness (shortest paths passing through).
- Remove highest betweenness edge, recording the state
- Repeat until no edges remain, maximizing modularity on the states.
- Connected components are the communities.

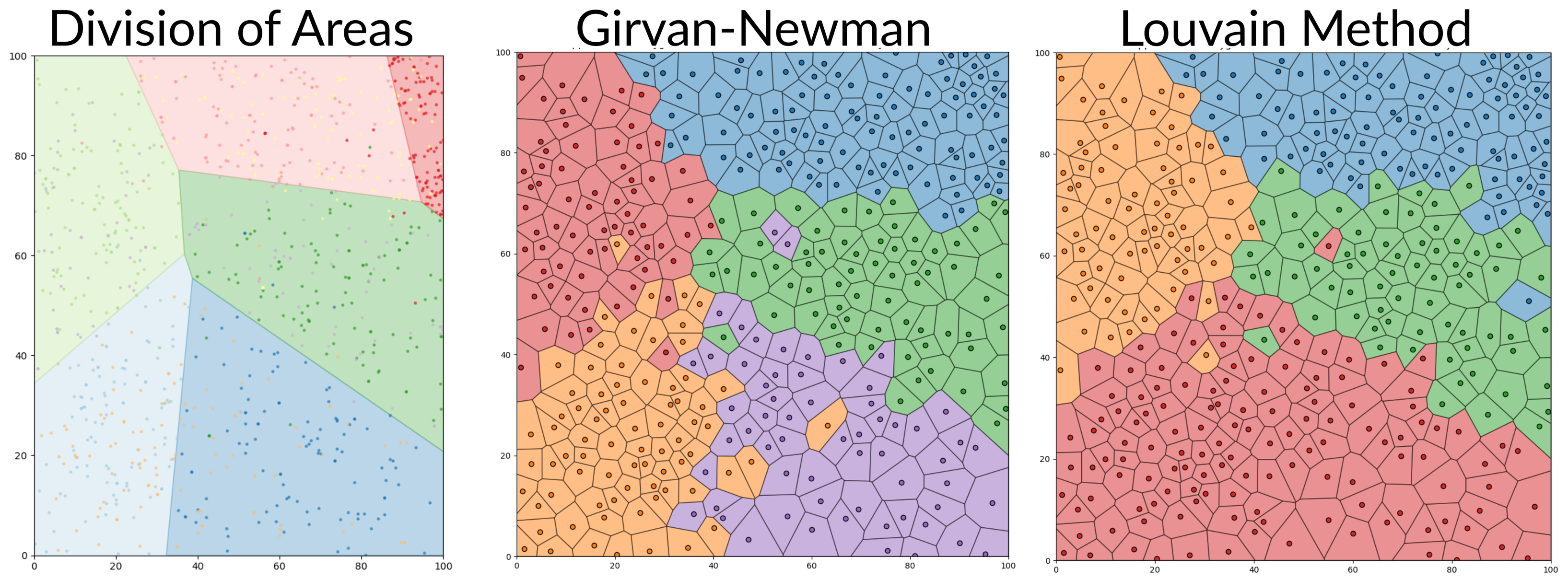


Louvain Method:



- Assign each node to its own community.
- Try adding each node to adjacent communities, calculating modularity.
- Move the node if modularity improves.
- Repeat until no modularity improvement is possible.

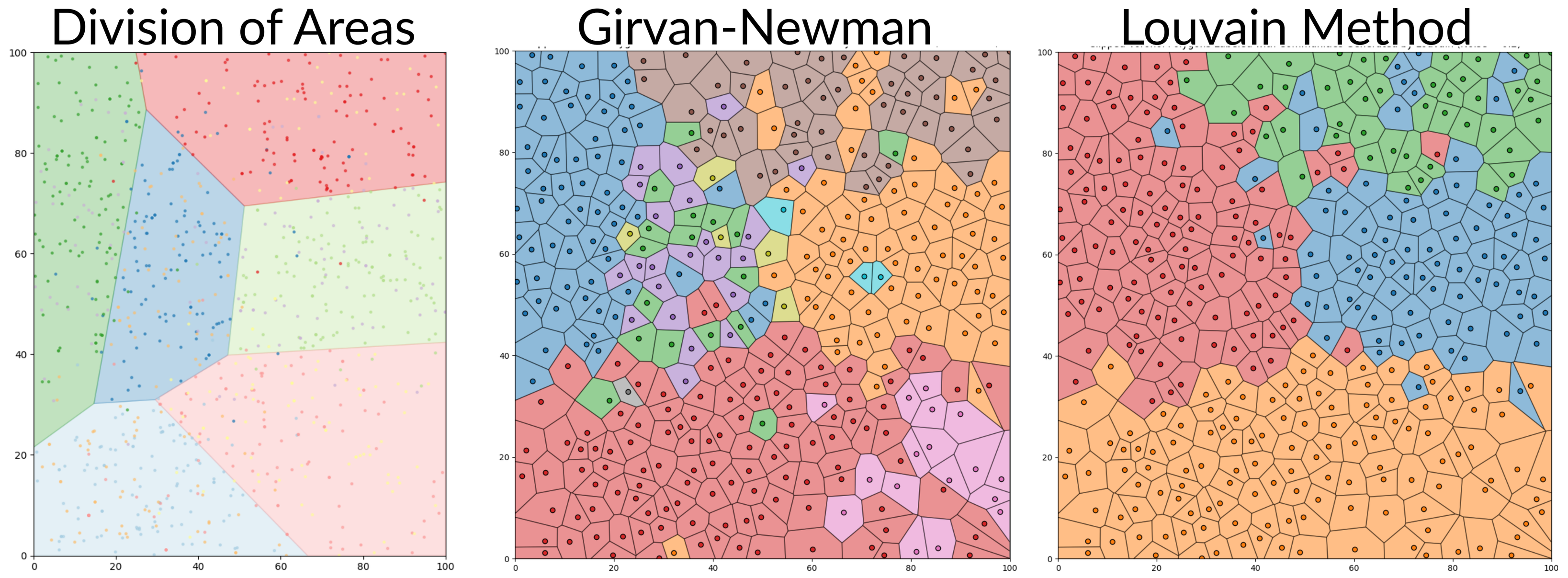
Results



Runtime	Time(s)
GN	19592.44
LV	557.48

Table 1.Noise = 10%

The Girvan-Newman algorithm detects distinct areas well, while the Louvain method merges neighboring regions (the bottom two). However, Girvan-Newman is nearly 39 times slower (19,592s vs. 557s) for 2,925 edges in this case.



Runtime	Time(s)
GN	3235.12
LV	335.55

Table 2.Noise = 20%

When we allow objects move randomly by 20%, the Girvan-Newman algorithm can detect communities at correct locations (with rougher boundaries), while the Louvain method tends to merge communities more aggressively but maintains clear boundary. The time consumption remains high, with a ratio of 10:1

Results Analysis

- **Community Detection:**
  - Girvan-Newman is more sensitive to the movement of objects in trajectories by detecting high-betweenness edges.
  - Louvain, optimizing modularity, tends to merge more communities by prioritizing dense internal connections, yet it maintains clear boundaries.
- **Computational Efficiency:**
  - Girvan-Newman is much slower (19,592s vs. 557s) as it repeatedly computes betweenness, requiring costly shortest-path calculations ($\mathcal{O}(|E||V|^2)$).
  - Louvain method, using a greedy heuristic, quickly merges nodes and aggregates communities, making it much faster ($\mathcal{O}(|V|\log|V|)$).
- **Summary of Trade-offs:**
  - Girvan-Newman provides finer structural insights but is computationally expensive.
  - Louvain is efficient but may over-merge communities due to its modularity-based approach.

Can We Retain Girvan-Newman's Accuracy with Louvain's Speed? (Future Work)

Conjecture Approch:

- Start with all nodes assigned to one large community.
- For each community, split it into two using betweenness (Girvan-Newman), and apply Louvain to adjust the two part.
- Check modularity. If modularity improves, keep the split; otherwise, retain the community.
- Continue until no further splitting or merging is possible.

Why This Might Work:

- **Targeted Splitting:** Betweenness identifies critical edges for a meaningful split, and after division, no further edge removal or betweenness recalculation is needed, reducing redundancy.
- **Louvain Boundary Refinement:** Since Louvain identifies clear boundaries, using it after splitting enhances boundary clarity.
- **Controlled Adjustments:** Louvain is applied only to the two newly split communities each time, making unintended merges unlikely.

References

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, October 2008. ISSN 1742-5468. doi: 10.1088/1742-5468/2008/10/p10008. URL http://dx.doi.org/10.1088/1742-5468/2008/10/P10008.

[2] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6), December 2004. ISSN 1550-2376. doi: 10.1103/physreve.70.066111. URL http://dx.doi.org/10.1103/PhysRevE.70.066111.

[3] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, June 2002. ISSN 1091-6490. doi: 10.1073/pnas.122653799. URL http://dx.doi.org/10.1073/pnas.122653799.

[4] Diansheng Guo, Hai Jin, Peng Gao, and Xi Zhu. Detecting spatial community structure in movements. 32(7):1326–1347. ISSN 1365-8816, 1362-3087. doi: 10.1080/13658816.2018.1434889. URL https://www.tandfonline.com/doi/full/10.1080/13658816.2018.1434889.

[5] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), February 2004. ISSN 1550-2376. doi: 10.1103/physreve.69.026113. URL http://dx.doi.org/10.1103/PhysRevE.69.026113.