Community Detection - Louvain Algorithm

Finding Densely Connected Groups in Networks

Communities

Densely connected groups with sparse inter-group connections

Modularity Optimization

Measure of community structure quality

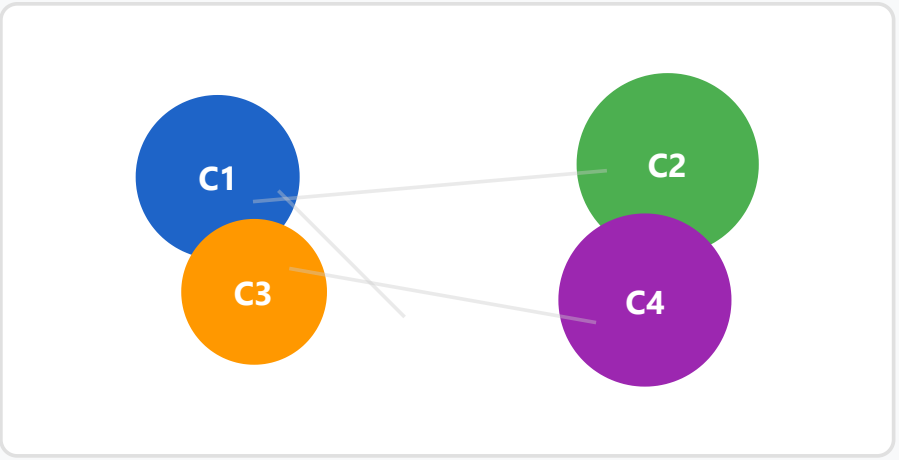
$$Q = \Sigma [e_{in} - (d_{tot}/2m)^2]$$

Two-Phase Iterative Approach

1 Local Phase

Move nodes to neighboring communities to maximize modularity gain

Community Structure Visualization



Key Advantages

- Fast & Scalable
- Hierarchical
- Large Networks
- High Quality

Applications

2

Global Phase

Aggregate communities into super-nodes, build new network

🔄 Iterate until modularity cannot be improved

Social Networks

Protein Interactions

Web Analysis

Citation Networks

Step 0: Initial State

2D Vector Input and Graph Construction

Input: 2D Vectors

```
v0:  (1.0, 1.5)
v1:  (1.2, 1.8)
v2:  (1.5, 1.3)
v3:  (5.0, 5.2)
v4:  (5.3, 5.5)
v5:  (5.1, 4.9)
```

Connection Criteria: Euclidean distance ≤ 1.0

Generated Edges:

• (v0, v1): 0.36 • (v0, v2): 0.54

• (v1, v2): 0.58 • (v3, v4): 0.42

• (v3, v5): 0.36 • (v4, v5): 0.63

Total Edges (m): 6

Initial State

Q = 0.000

Each node forms its own community.

Step 1: Move Node v0

v0 → C1 (ΔQ = +0.0833)

🔍 Node v0 Analysis

Neighbors: v1(C1), v2(C2)

Move Simulation:

• To C1: ΔQ = +0.0833 ✓

• To C2: ΔQ = +0.0694

Move v0 to C1

v0 → C1

Q = 0.083

C1={v0, v1}, C2={v2}, C3={v3}, C4={v4}, C5={v5}

Step 2: Check Node v1

v1 stays (already optimal)

🔍 Node v1 Analysis

Neighbors: v0(C1), v2(C2)

Move Simulation:

• To C2: ΔQ ≈ 0

Stay in current position

v1 stays

Q = 0.083

No change

Step 3: Move Node v2

v2 → C1 (ΔQ = +0.167)

🔍 Node v2 Analysis

Neighbors: v0(C1), v1(C1)

Move Simulation:

• To C1: ΔQ = +0.167 ✓

Move v2 to C1

Left cluster complete!

v2 → C1

Q = 0.250

C1={v0, v1, v2}, C3={v3}, C4={v4}, C5={v5}

Step 4: Move Node v3

v3 → C4 (ΔQ = +0.0833)

🔍 Node v3 Analysis

Neighbors: v4(C4), v5(C5)

Move Simulation:

• To C4: ΔQ = +0.0833 ✓

• To C5: ΔQ = +0.0694

Move v3 to C4

v3 → C4

Q = 0.333

C1={v0, v1, v2}, C4={v3, v4}, C5={v5}

Step 5: Check Node v4

v4 stays

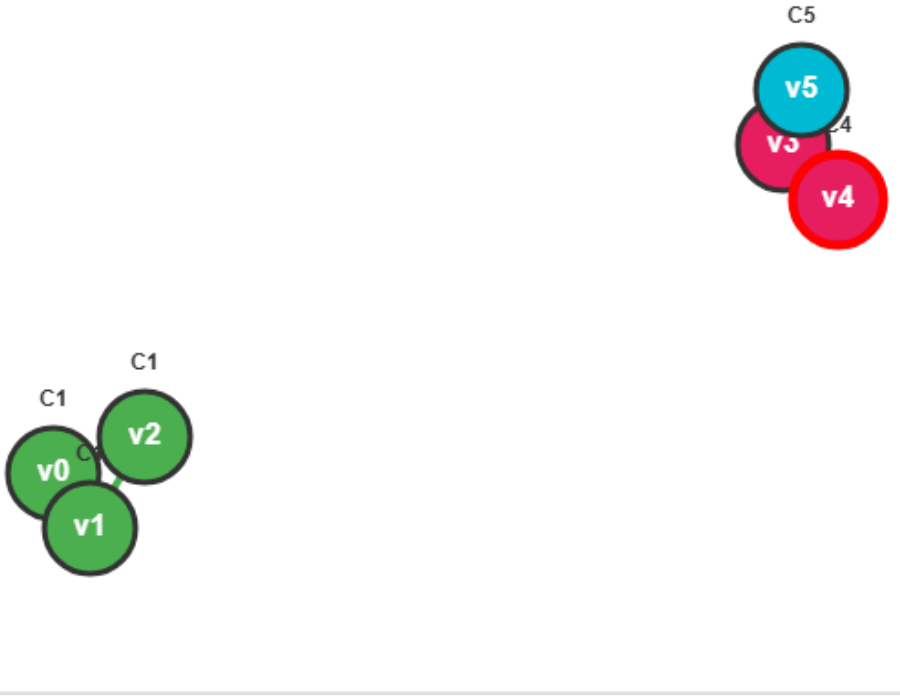
🔍 Node v4 Analysis

Neighbors: v3(C4), v5(C5)

Move Simulation:

• To C5: ΔQ ≈ 0

Stay in current position



v4 stays

Q = 0.333

No change

Step 6: Phase 1 Complete

v5 → C4 (ΔQ = +0.167)

🔍 Node v5 Analysis

Neighbors: v3(C4), v4(C4)

Move Simulation:

• To C4: ΔQ = +0.167 ✓

Move v5 to C4

Phase 1 Complete!

Phase 1 Complete

Q = 0.500

Final: C1={v0,v1,v2}, C4={v3,v4,v5}

Step 7: Phase 2 - Final Result

Super-node Creation and Algorithm Convergence

Super-nodes

S1 = {v0, v1, v2}

- Internal edges: 3
- Degree: 6

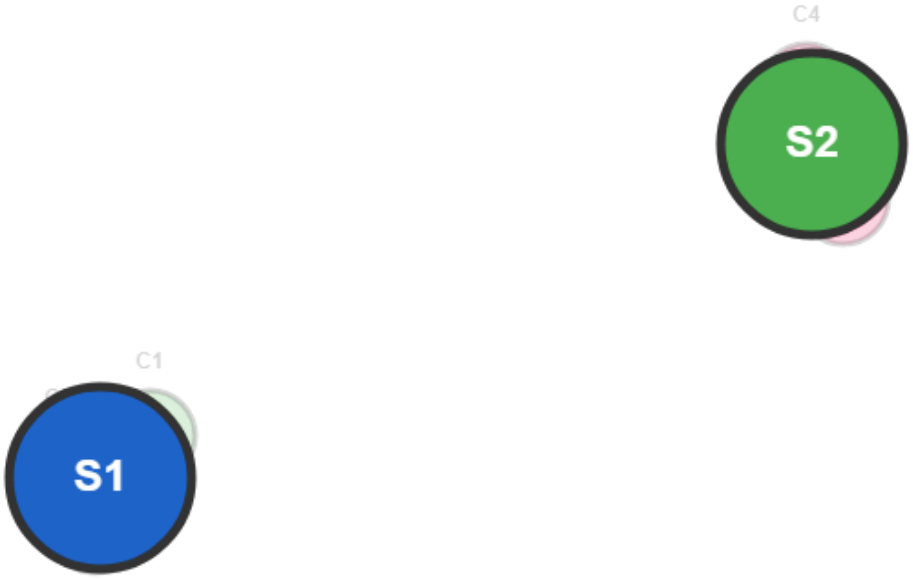
S2 = {v3, v4, v5}

- Internal edges: 3
- Degree: 6

New Graph:

- Nodes: 2
- Edges: 0

✓ Algorithm Terminated



Final Result

Q = 0.500

2 communities found!

No further improvement possible

📊 Algorithm Summary

Phase 1 Results

- Node moves: **4**
- Initial communities: **6**
- Final communities: **2**
- Modularity: **0.000** → **0.500**

Phase 2 Results

- Super-nodes: **2**
- Hierarchy levels: **2**
- Inter-community edges: **0**
- Status: **Converged** ✓