Network 4 – The Class Network

Giorgio Ricchiuti
www.grarchive.net
giorgio.ricchiuti@unifi.it

CompEc

From Theory to Your Own Network

What we have done so far:

- Centrality measures on synthetic networks
- Community detection on the Karate Club (Zachary, 1977)
- All on *someone else's* data

Today:

- We apply the **same tools** to a real network
- The nodes are **you**
- The edges are your **group preferences** for the Problem Set

The data

- You filled in a Google Form
- You indicated up to **4 classmates** you would like to work with
- This gives us a **directed** network: $i \to j$ means "i chose j"
- Node attributes: **Python skill** and **expected hours**

Research Questions

Before running the code, let us state what we want to find out:

1. **Structure**: is this a connected network? How dense is it?
   → basic statistics, degree distribution
2. **Reciprocity**: if i chose j, did j also choose i?
   → a measure of mutual preference
3. **Influence**: who are the most "popular" or "central" students?
   → degree, betweenness, closeness, PageRank, Katz
4. **Groups**: do natural study groups emerge from the data?
   → community detection (Louvain)
5. **Attributes**: is Python skill clustered within communities?
   → node attribute analysis

Building the Network

From form to graph:

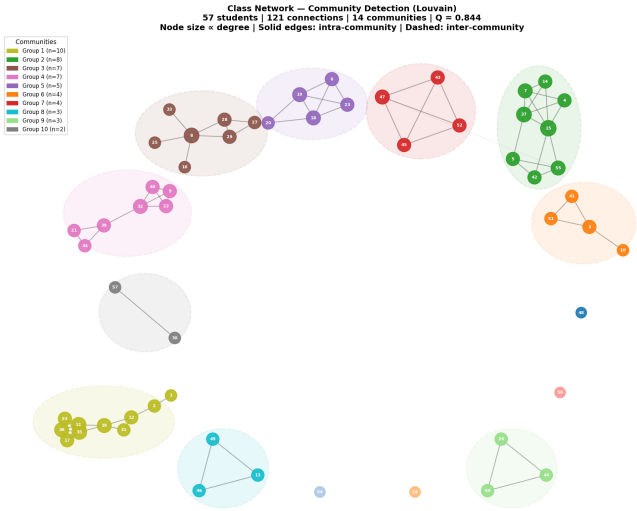
1. Load the Excel file with `pandas`
2. **Anonymise**: replace student IDs with numbers $1, 2, \ldots, 57$
3. For each student i and each preference j: add directed edge $i \to j$
4. Build a `DiGraph` and an undirected version for centrality and community detection

Network at a glance

Students (nodes)	57
Connections (edges)	121
Average degree	2.12
Reciprocity	0.760
Connected components	13
Largest component	12 students

Reciprocity = 0.76: 76% of choices are mutual — strong evidence of pre-existing friendship or study groups.

The Network


Class Network — Community Detection (Louvain)
57 students | 121 connections | 14 communities | Q = 0.844
Node size ∝ degree | Solid edges: intra-community | Dashed: inter-community
Communities
Group 1 (n=10)
Group 2 (n=8)
Group 3 (n=7)
Group 4 (n=7)
Group 5 (n=5)
Group 6 (n=4)
Group 7 (n=4)
Group 8 (n=3)
Group 9 (n=3)
Group 10 (n=2)

Why is the Network Fragmented?

The network has **13 connected components** — students are not all reachable from one another.

Possible explanations:

- Students chose only from their existing social circle
- Some students filled in the form late or incompletely
- No "bridges" between certain social groups

Component sizes:
12, 10, 7, 7, 5, 4, 3, 3, 2, 1, 1, 1, 1

Implications

- **Closeness** centrality is computed within each component — cross-component distances are ∞
- **Betweenness** is zero for nodes in singleton components
- The 4 **isolated nodes** (singletons) are students who gave preferences only outside the class respondents, or gave none at all

Centrality Measures – Recap

We compute **five centrality measures** on the undirected network:

Degree Centrality

$$C_D(i) = \frac{k_i}{n-1}$$

Who has the most connections?

Betweenness Centrality

$$C_B(i) = \sum_{s \neq i \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}$$

Who is the most important bridge?

Katz Centrality

$$C_K(i) = \sum_{k=1}^{\infty} \alpha^k (A^k)_{ji}$$

Like eigenvector, but works on disconnected graphs

Closeness Centrality

$$C_C(i) = \frac{n-1}{\sum_j d(i,j)}$$

Who can reach everyone most quickly?

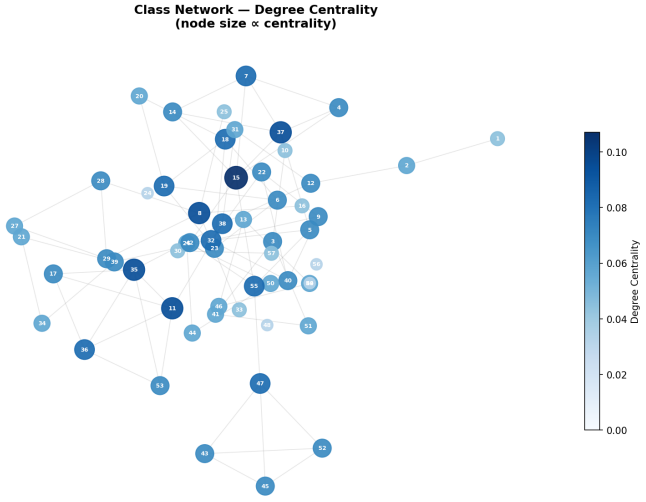
PageRank

$$PR(i) = \frac{1-\alpha}{n} + \alpha \sum_{j \to i} \frac{PR(j)}{k_j^{out}}$$

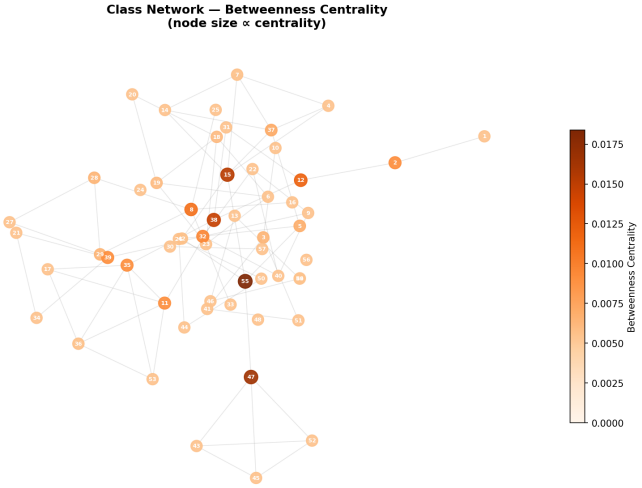
Who is chosen by well-connected students?

Note: Eigenvector centrality is undefined for disconnected graphs $\Rightarrow$ we use **Katz** instead.

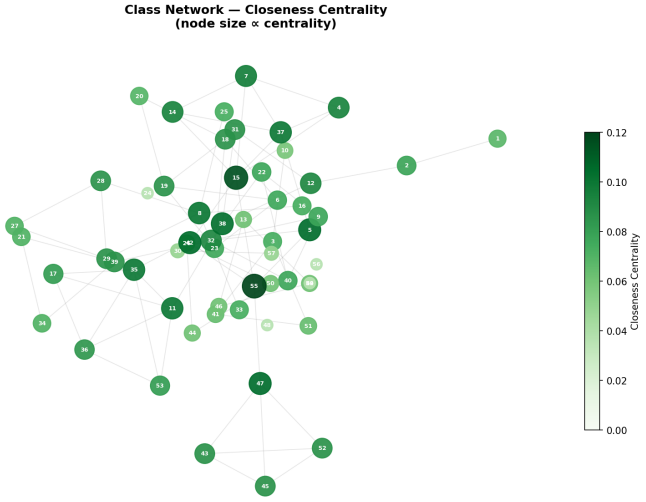
Centrality – Degree



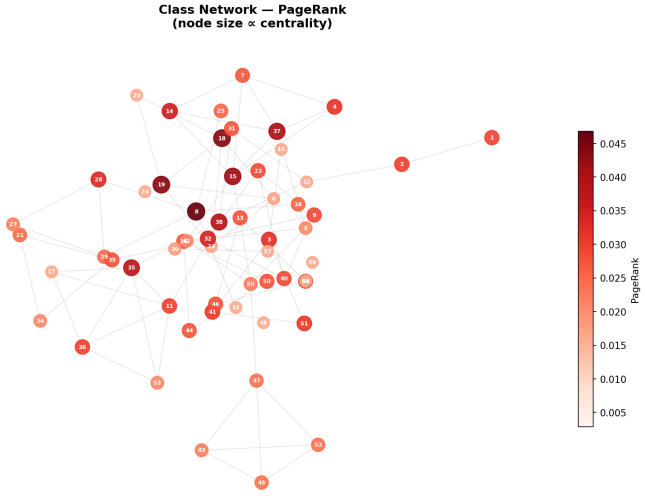
Centrality – Betweenness



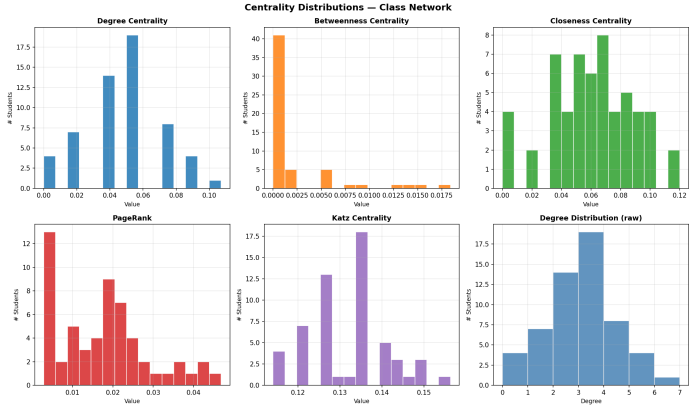
Centrality – Closeness



Centrality – PageRank



Centrality – Distributions



Centrality – Discussion

Key findings:

- **Degree**: student 15 is the most connected (6 links)
- **Betweenness**: student 55 is the main bridge between components — removing this node would disconnect parts of the network
- **PageRank**: students 8, 18, 19 are chosen by well-connected peers
- High betweenness $\neq$ high degree: **different roles** in the network

Roles in social networks

- High **degree** $\rightarrow$ popular, many friends
- High **betweenness** $\rightarrow$ broker, connects otherwise distant groups
- High **closeness** $\rightarrow$ fast information diffusion
- High **PageRank** $\rightarrow$ endorsed by influential peers
- High **Katz** $\rightarrow$ connected to highly connected nodes (like eigenvector)

Community Detection – Louvain

We apply the **Louvain algorithm** to the undirected network.

Results:

- **14 communities** detected
- Modularity $Q = 0.844$ — extremely high
- Largest community: 10 students
- 4 singleton communities (isolated students)

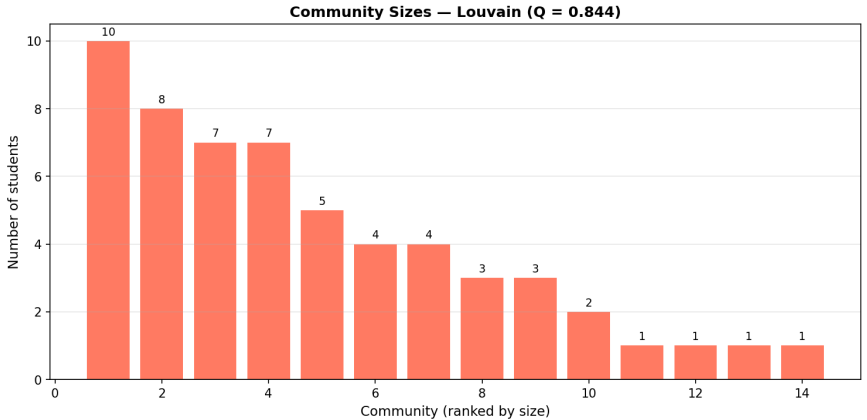
Interpretation of $Q = 0.844$

Well above the 0.3 threshold.
The class is **very strongly modular**: students chose almost exclusively within their own social group.

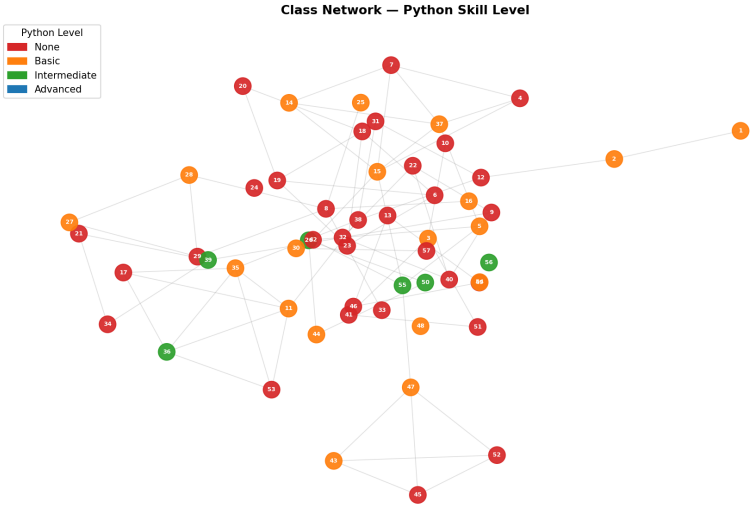
Largest communities:

Community	Size	Students
1	10	1,2,11,12,17,31,35,36,38,53
2	8	4,5,7,14,15,37,42,55
3	7	8,16,25,27,28,29,33
4	7	9,21,22,32,34,39,40
5	5	6,18,19,20,23

Community Sizes



Python Skill Across the Network



Python Skill – Discussion

What we observe:

- Python skill is **not uniformly distributed** across communities
- Most students are "None" or "Basic"
- Students with **Intermediate** skill tend to appear in the same communities
- No **Advanced** users in the class

Why does this matter?

- If skill is clustered *within* communities, some groups may struggle with the Problem Set
- A **heterogeneous** group (mixed skills) might produce better learning outcomes
- This is a network-based argument for *optimal group assignment*: the professor can use centrality and community information to form balanced groups

Python: What We Will Do

1. Load the anonymised Excel file and build the network
2. Compute **basic statistics**: density, reciprocity, components
3. Compute **five centrality measures** and visualise them as network plots (node size $\propto$ centrality, Kamada–Kawai layout)
4. Plot the **distribution** of each centrality measure
5. Run **Louvain** community detection and compute modularity Q
6. Visualise by **community** and by **Python skill**

Key functions

- `nx.DiGraph()` / `.to_undirected()`
- `nx.kamada_kawai_layout()`
- `nx.reciprocity()`
- `nx.degree_centrality()`
- `nx.betweenness_centrality()`
- `nx.closeness_centrality()`
- `nx.pagerank()`
- `nx.katz_centrality_numpy()`
- `community_louvain.best_partition()`
- `nx.community.modularity()`

Python: Build the Network

```
import pandas as pd, networkx as nx

df = pd.read_excel('class_network_data.xlsx')
# columns: student_id, from_student, to_student, ...

# Load edge list
df_edges = pd.read_excel('class_network_edgelist.xlsx')

G = nx.DiGraph()
G.add_nodes_from(range(1, 58))
G.add_edges_from(zip(df_edges['from_student'],
                     df_edges['to_student']))
G_und = G.to_undirected()

print(f"Nodes: {G.number_of_nodes()}")
print(f"Edges: {G.number_of_edges()}")
```

Python: Centrality and Layout

```
# Kamada-Kawai layout (better than spring for this network)
pos = nx.kamada_kawai_layout(G_und)

# Centrality measures
degree_c    = nx.degree_centrality(G_und)
betweenness = nx.betweenness_centrality(G_und, normalized=True)
closeness   = nx.closeness_centrality(G_und)
pagerank    = nx.pagerank(G, alpha=0.85)
katz        = nx.katz_centrality_numpy(G_und, alpha=0.05)

# Visualise: node size proportional to centrality
node_sizes = [150 + 4000 * degree_c[n] for n in G_und.nodes()]
nx.draw_networkx(G_und, pos=pos, node_size=node_sizes, ...)
```

Python: Community Detection

```
import community as community_louvain
from collections import defaultdict

partition   = community_louvain.best_partition(G_und,
                                               random_state=42)
comm_groups = defaultdict(set)
for node, c in partition.items():
    comm_groups[c].add(node)
comm_list = list(comm_groups.values())
Q = nx.community.modularity(G_und, comm_list)

print(f"Communities: {len(comm_list)},  Q = {Q:.4f}")

# Colour nodes by community
palette     = ['#1f77b4','#ff7f0e','#2ca02c','#d62728', ...]
node_colors = [palette[partition[n] % len(palette)]
```

Two Benchmark Network Models

Erdős–Rényi Random Graph $G(n, p)$

- n nodes; each pair connected with probability p, independently
- Edges placed **at random** — no structure
- Degree distribution: **Poisson** (bell-shaped, around $\bar{k} = p(n-1)$)
- Low clustering, short paths
- Benchmark for the *absence* of social structure

Watts–Strogatz Small World $G(n, k, p)$

- Start from a **regular ring**: each node connected to k nearest neighbours
- Rewire each edge with probability p
- Result: **high clustering** (like a lattice) *and* **short paths** (like random)
- The "six degrees of separation" phenomenon
- Realistic model for social and economic networks

Key question: which model is the class network closest to?

Small World: Intuition

The signature of a small-world network:

- **High clustering** $C \gg C_{\text{random}}$: your friends tend to know each other
- **Short average path** $L \approx L_{\text{random}}$: despite local clustering, you can still reach everyone in few steps

Small World Index:

$$\sigma = \frac{C/C_{\text{random}}}{L/L_{\text{random}}} \gg 1$$

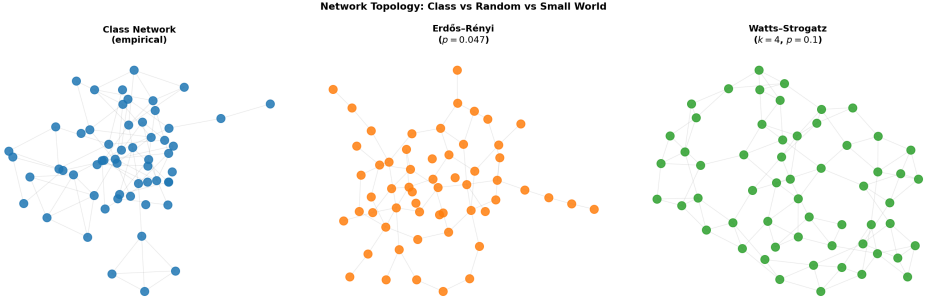
$\sigma > 1$ is the standard criterion. The larger σ, the more "small-worldish" the network.

Examples: neural networks, power grids, social networks, financial contagion networks.

Watts–Strogatz construction

1. Ring lattice: n nodes, each linked to k neighbours
2. For each edge: rewire with probability p
3. $p = 0$: pure lattice (high C, high L)
4. $p = 1$: pure random (low C, low L)
5. $p \approx 0.01$–0.1: **small world** (high C, low L)

Comparing Network Topologies



Class Network vs Benchmarks

	Class	Erdős–Rényi	Watts–Strogatz
Clustering C	**0.618**	0.036	0.373
Avg path L (LCC)	**2.23**	3.98	4.11
Components	**13**	$\sim$1	$\sim$1
C/C_{ER}	**17.3**$\times$	1$\times$	10.4$\times$
L/L_{ER}	**0.56**	1	1.03
σ	**30.8**	1	$\sim$10

L on LCC; C on full graph. ER and WS: averages over 100 simulations, $n = 57$, $p = 0.047$, $k = 4$, $p_{\text{rew}} = 0.1$.

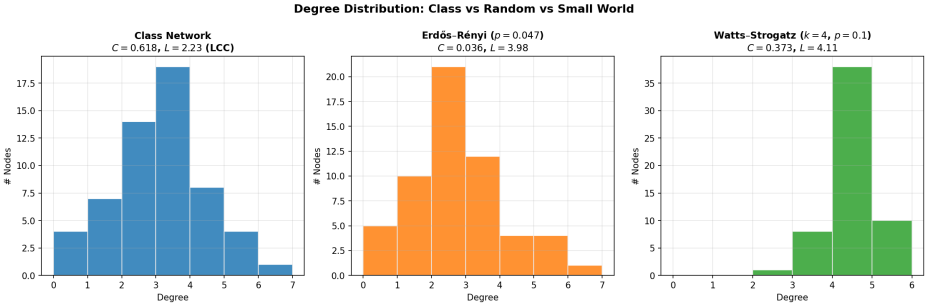
What we learn:

- Clustering 17$\times$ higher than random — friends of friends are friends
- Paths within each component very short ($L = 2.23$)
- $\sigma = 30.8 \gg 1$: **strongly small-world**
- But **13 components**: more fragmented than any benchmark

Interpretation

The class is a collection of **tight cliques** — more clustered than a small-world, but lacking the long-range links that would bridge them.

Degree Distributions: Class vs Benchmarks



What This Means for the Next Lecture

The standard BH model assumes:

- Agents interact with **everyone** (complete graph)
- All agents observe the **same price signal**
- No local structure, no communities

But real networks are different:

- **Clustered**: agents interact mostly within their group
- **Fragmented**: some agents are isolated from others
- **Small world**: local clustering + short global paths

Open questions for Wednesday

- What happens to BH dynamics when agents only observe **neighbours' strategies**?
- Does network topology **amplify or dampen** price fluctuations?
- Can **local herding** within communities trigger **global instability**?
- Does a **small-world** structure make markets more or less stable than a random graph?

Summary

What we found:

- The class network is **fragmented**: 13 separate components
- **Reciprocity** 0.76: preferences are mostly mutual
- Modularity $Q = 0.844$: extremely strong community structure
- Centrality reveals **different roles**: popular nodes, brokers, endorsed nodes
- Python skill is **unevenly distributed** across communities
- $\sigma = 30.8$: the class is **strongly small-world** but fragmented

Broader lesson:

- Network tools reveal **hidden structure** in real social data
- The same algorithms used on the Karate Club work on *your own* data
- Community detection has direct policy implications: group assignment, information diffusion, peer effects in learning
- Real networks are neither random nor regular — they are **small worlds**

Next lecture

Brock & Hommes model **on a network**: what happens to financial market dynamics when