Network Analysis:

The Hidden Structures behind the Webs We Weave

17-213 / 17-668

Structural Equivalence

Tuesday, October 3, 2023

Patrick Park & Bogdan Vasilescu

Carnegie Mellon University
School of Computer Science

S3D
Software and Societal Systems Department

2-min Quiz, on Canvas

Quick Recap – Last Thursday's Lecture

Big question: What is a social group?

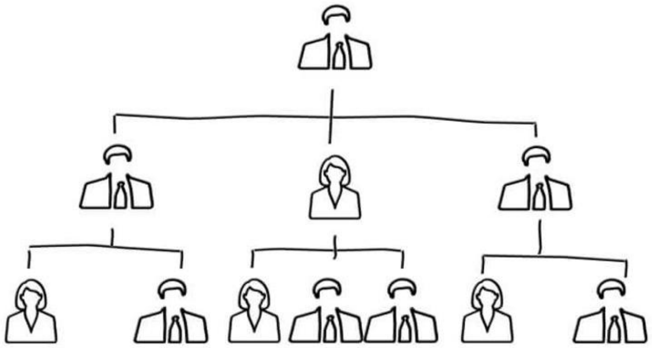
Criteria: How close or reachable are people? The focus underlying this way of approaching the question of groups is asking how two people are related through some notion of "distance"

Distance: How close is i to j?

- Attraction: i and j are brought closer by more ties around them (closure)
- Repulsion: i and j are pushed farther apart if their other neighbors are farther apart (fewer links between lumps of nodes) → Modularity, betweenness, etc.

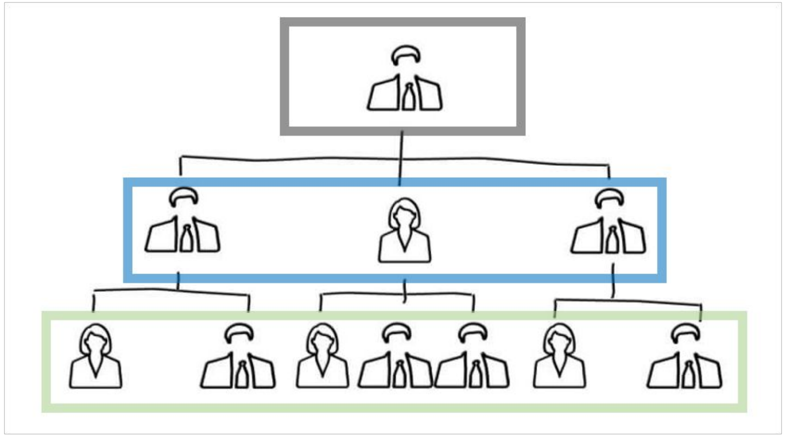
Structural Equivalence

Groups Based on Equivalent Positions



Position: Individuals who are similarly embedded in networks of relations

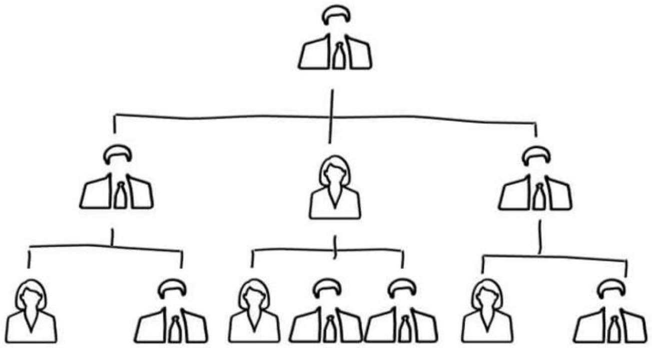
Groups Based on Equivalent Positions



Position: Individuals who are **similarly** embedded in networks of relations

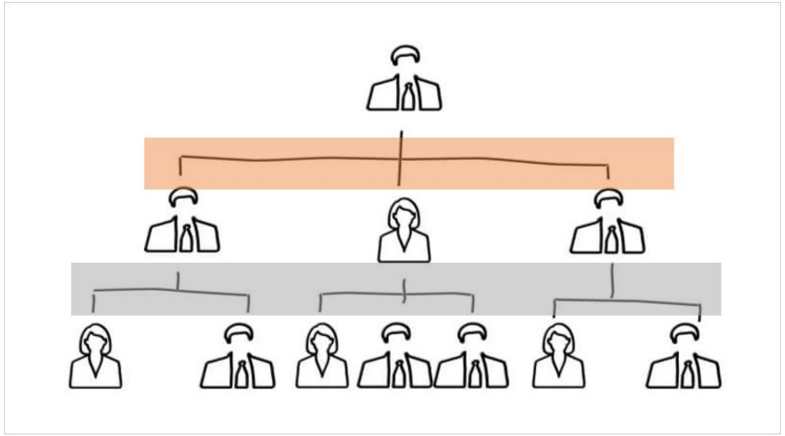
Similar relational pattern **does not** imply direct connection

Groups Based on Equivalent Positions



Role: Patterns of relations between positions

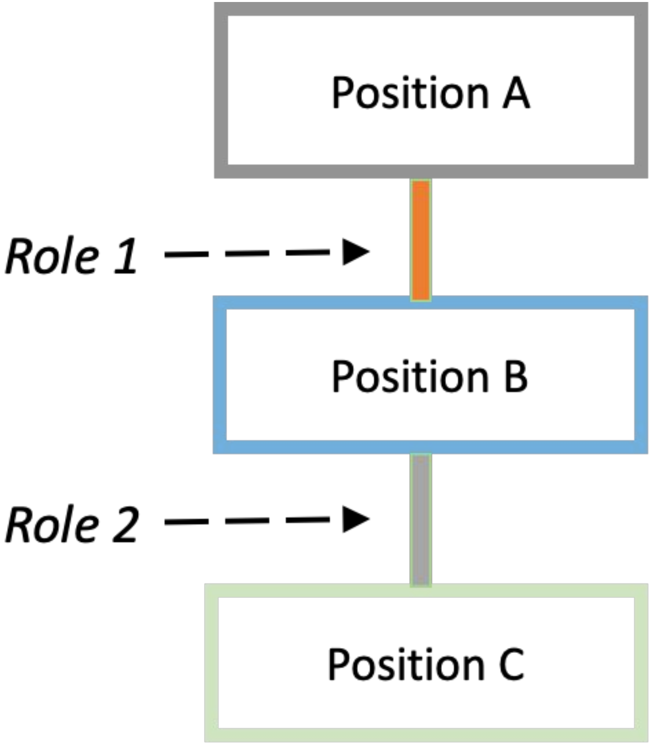
Groups Based on Equivalent Positions



Role: Patterns of relations between positions

Not just one relation between two positions, but the totality of relations in a network

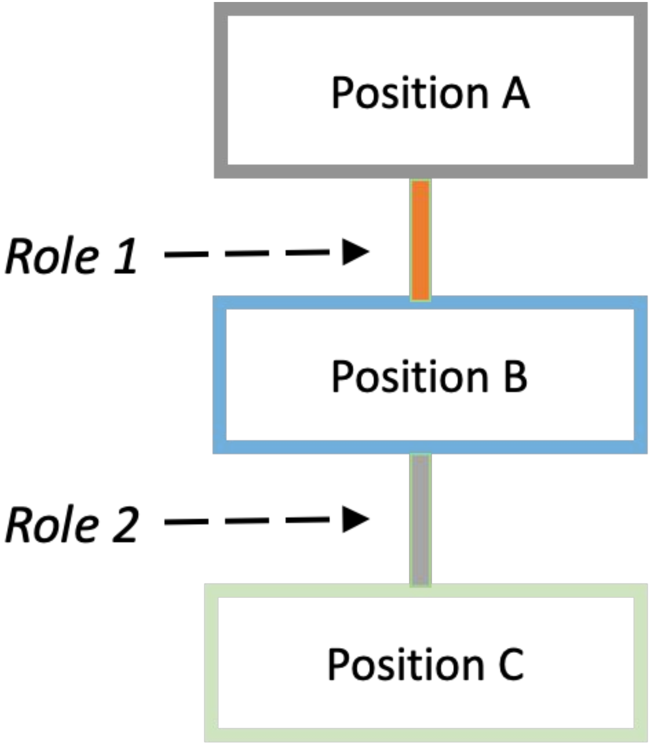
Groups Based on Equivalent Positions



Positional analysis aims to simplify the data into positions and roles

Example: People in Position A have *Role 1* in relation to Position B

Groups Based on Equivalent Positions

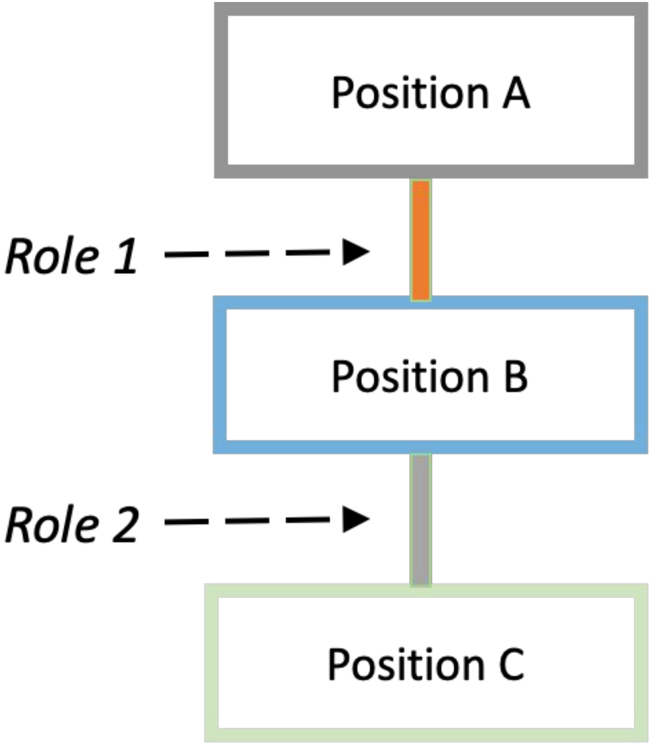


In the analysis of positions and roles, each position is a node and each role is a tie

Aim: Simplification/data reduction of the network

Block model is one method of data simplification into roles and positions

Groups Based on Equivalent Positions



1. Decide definition of “equivalence” that will be used to group nodes into same position

2. Since real networks often do not perfectly fit the formal definition of equivalence, measure the degree to which subset of nodes approach the definition

3. Represent the equivalence classes and equivalence relations

4. Assess adequacy

Different Definitions of Equivalence

Stringent

Structural Equivalence

Automorphic Equivalence

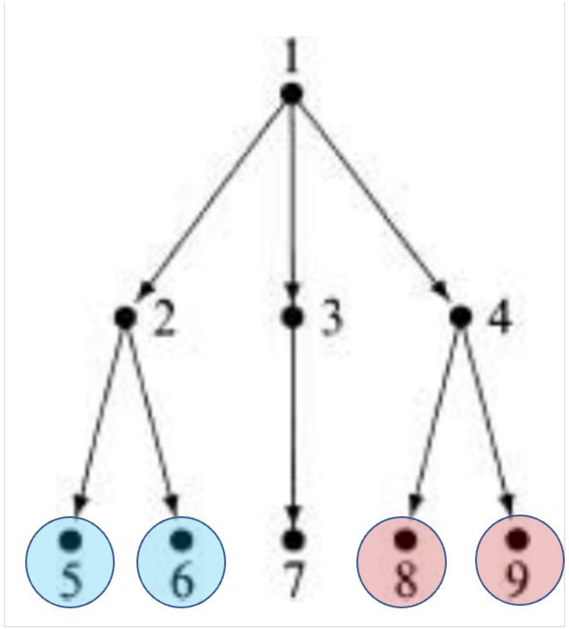
Isomorphic Equivalence

Regular Equivalence

…

General

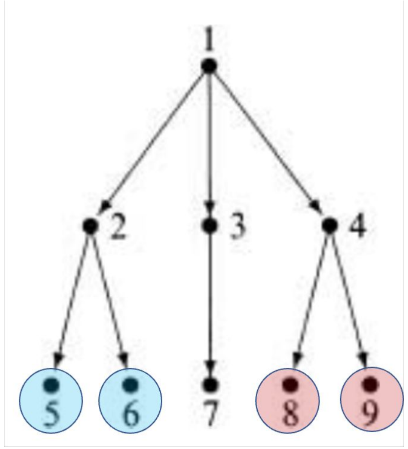
Structural Equivalence



Nodes that have incoming and outgoing ties to the same set of other nodes on all relationship types

{1}, {2}, {3}, {4}, {5,6}, {7}, {8,9}

Structural Equivalence



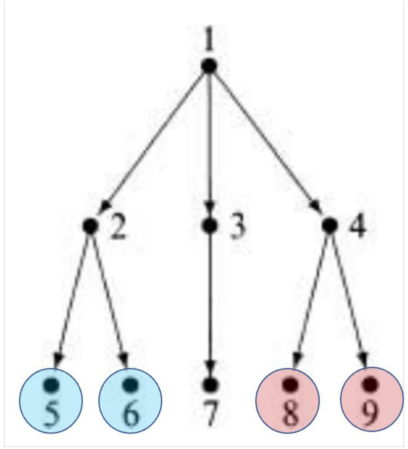
	2	5	6
2	0	1	1
5	0	0	0
6	0	0	0

In an adjacency matrix, structural equivalence means two nodes have the exact same row and column values

Example: 5 and 6 share exact same column vectors

→ incoming ties from same node

Structural Equivalence



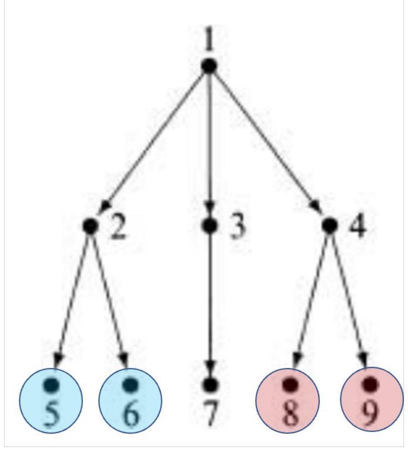
	2	5	6
2	0	1	1
5	0	0	0
6	0	0	0

In an adjacency matrix, structural equivalence means two nodes have the exact same row and column values

Example: 5 and 6 share exact same row vectors

→ outgoing ties to same node(s)

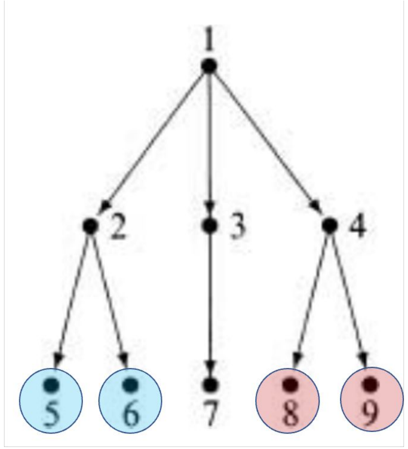
Structural Equivalence



	2	5	6
2	0	200	200
5	0	0	0
6	0	0	0

If network is weighted, edge weights must be identical as well for two nodes to be structurally equivalent

Structural Equivalence



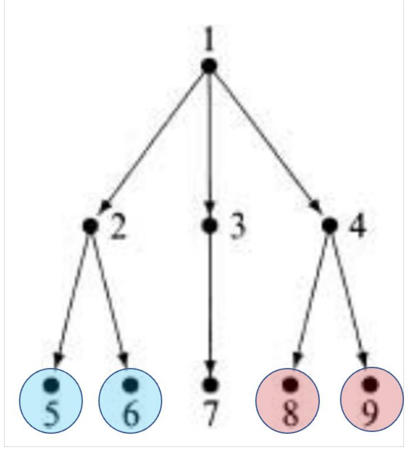
	2	5	6
2	0	200	200
5	0	0	0
6	0	0	0

In reality, nodes that satisfy these conditions are rare

Alternative: How close do these nodes approach the formal definition of structural equivalence?

→ Measured by Euclidean Distance or Correlation

Structural Equivalence



	2	5	6
2	0	200	200
5	0	0	0
6	0	0	0

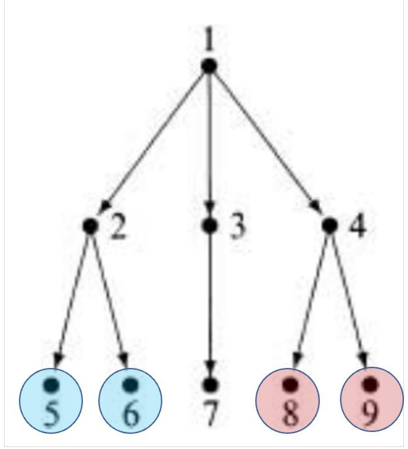
Euclidean distance

$$d_{ij} = \sqrt{\sum_{k=1}^{g} \left[(x_{ik} - x_{jk})^2 + (x_{ki} - x_{kj})^2 \right]}$$

Nodes 5 and 6 have 0 distance

Perfectly structurally equivalent

Structural Equivalence



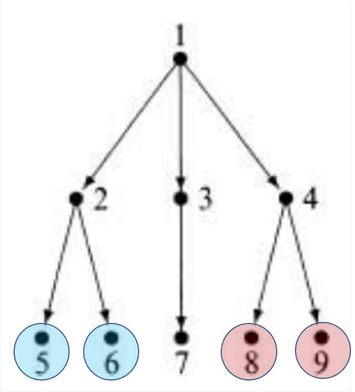
	2	5	6
2	0	200	200
5	0	0	0
6	0	0	0

Euclidean distance

$$d_{ij} = \sqrt{\sum_{r=1}^{R} \sum_{k=1}^{g} [(x_{ikr} - x_{jkr})^2 + (x_{kir} - x_{kjr})^2]}$$

Measurable across a set, *R*, of multiple relations (e.g., R={mentor, friendship, department}

Structural Equivalence



	2	5	6
2	0	200	200
5	0	0	0
6	0	0	0

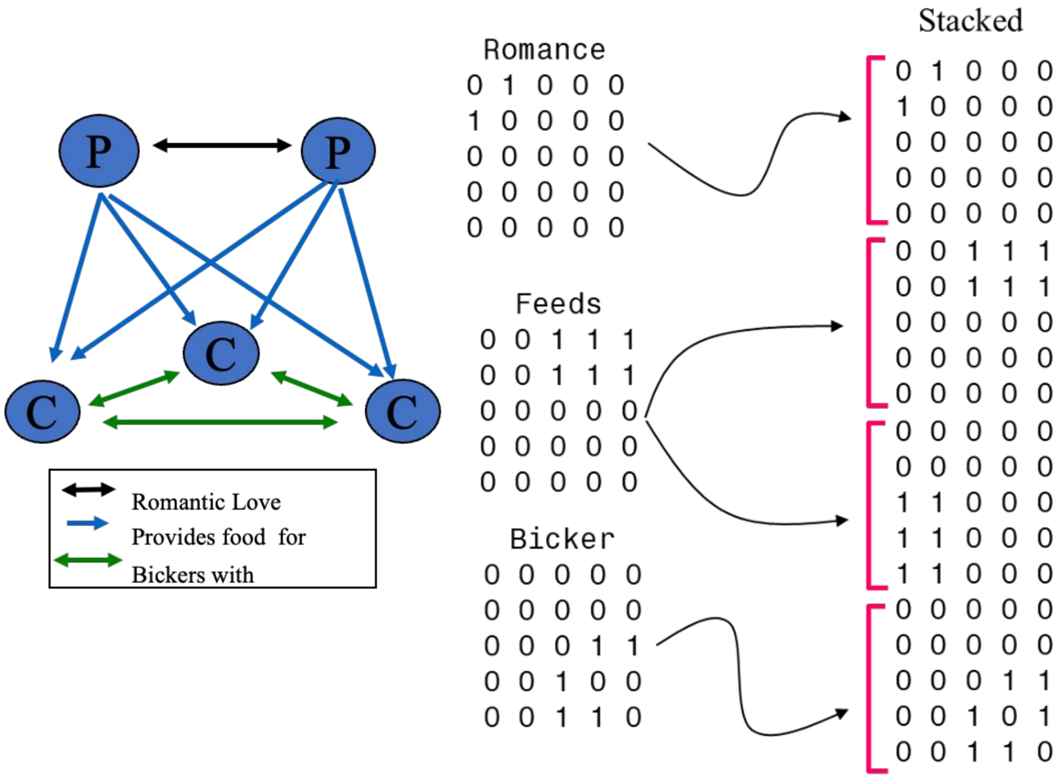
Pearson correlation coefficient

$$r_{ij} = \frac{\sum(x_{ki} - \overline{x_{\cdot i}})(x_{kj} - \overline{x_{\cdot j}}) + \sum(x_{ik} - \overline{x_{i\cdot}})(x_{jk} - \overline{x_{j\cdot}})}{\sqrt{\sum(x_{ki} - \overline{x_{\cdot i}})^2 + \sum(x_{ik} - \overline{x_{i\cdot}})^2}\sqrt{\sum(x_{kj} - \overline{x_{\cdot j}})^2 + \sum(x_{jk} - \overline{x_{j\cdot}})^2}}$$

Correlation of two nodes' column vectors and row vectors

Larger value indicates higher equivalence

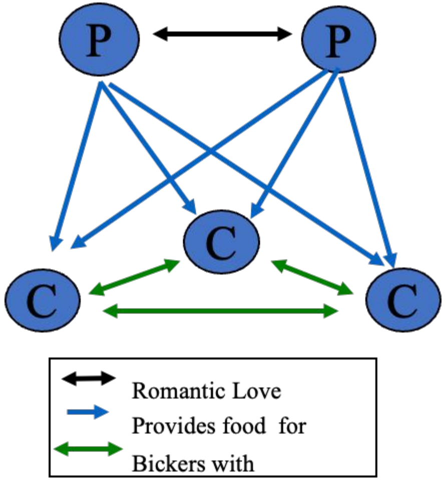
Structural Equivalence



Measure the similarity in a pair's connections to all other nodes

Examples:
Cosine similarity
Pearson correlation
Euclidean distance

Structural Equivalence



Romance

0	1	0	0	0
1	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Feeds

0	0	1	1	1
0	0	1	1	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Bicker

0	0	0	0	0
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0

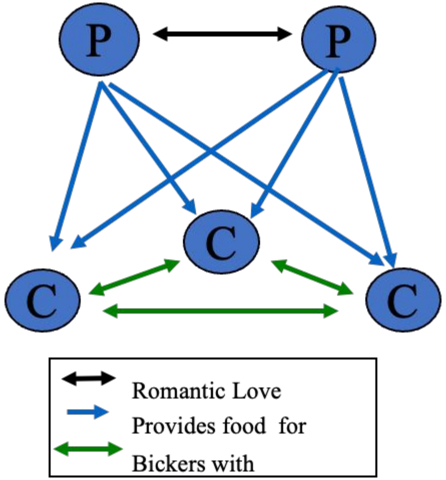
Stacked

0	1	0	0	0
1	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	1	1	1
0	0	1	1	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0

Euclidian Distance Matrix

	P	P	C	C	C
P		1.41	3.00	3.00	3.00
P	1.41		3.00	3.00	3.00
C	3.00	3.00		1.41	1.41
C	3.00	3.00	1.41		1.41
C	3.00	3.00	1.41	1.41	

Structural Equivalence



Romance

0	1	0	0	0
1	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Feeds

0	0	1	1	1
0	0	1	1	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Bicker

0	0	0	0	0
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0

Stacked

0	1	0	0	0
1	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	1	1	1
0	0	1	1	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0

Euclidian Distance Matrix

	P	P	C	C	C
P		1.41	3.00	3.00	3.00
P	1.41		3.00	3.00	3.00
C	3.00	3.00		1.41	1.41
C	3.00	3.00	1.41		1.41
C	3.00	3.00	1.41	1.41	

Objective: group the nodes that have similar distances

Structural Equivalence

	P	P	C	C	C
P		1.41	3.00	3.00	3.00
P	1.41		3.00	3.00	3.00
C	3.00	3.00		1.41	1.41
C	3.00	3.00	1.41		1.41
C	3.00	3.00	1.41	1.41	

Partition the Euclidean distance matrix (Pearson correlation matrix) into blocks where nodes in the same block have similar distances to the rest of the nodes

CONCOR:

CONvergence of iterated **COR**relations

Structural Equivalence

Euclidean Distance Matrix

	P	P	C	C	C
P		1.41	3.00	3.00	3.00
P	1.41		3.00	3.00	3.00
C	3.00	3.00		1.41	1.41
C	3.00	3.00	1.41		1.41
C	3.00	3.00	1.41	1.41	

Compute correlation coefficient of pairs of column vectors

Correlation Matrix C1

	P	P	C	C	C
P		$r_{p,p}$	$r_{p,c}$	$r_{p,c}$	$r_{p,c}$
P	$r_{p,p}$		$r_{p,c}$	$r_{p,c}$	$r_{p,c}$
C	$r_{p,p}$	$r_{p,c}$		$r_{c,c}$	$r_{c,c}$
C	$r_{p,c}$	$r_{p,c}$	$r_{c,c}$		$r_{c,c}$
C	$r_{p,c}$	$r_{p,c}$	$r_{c,c}$	$r_{c,c}$	

Structural Equivalence

Correlation Matrix C1

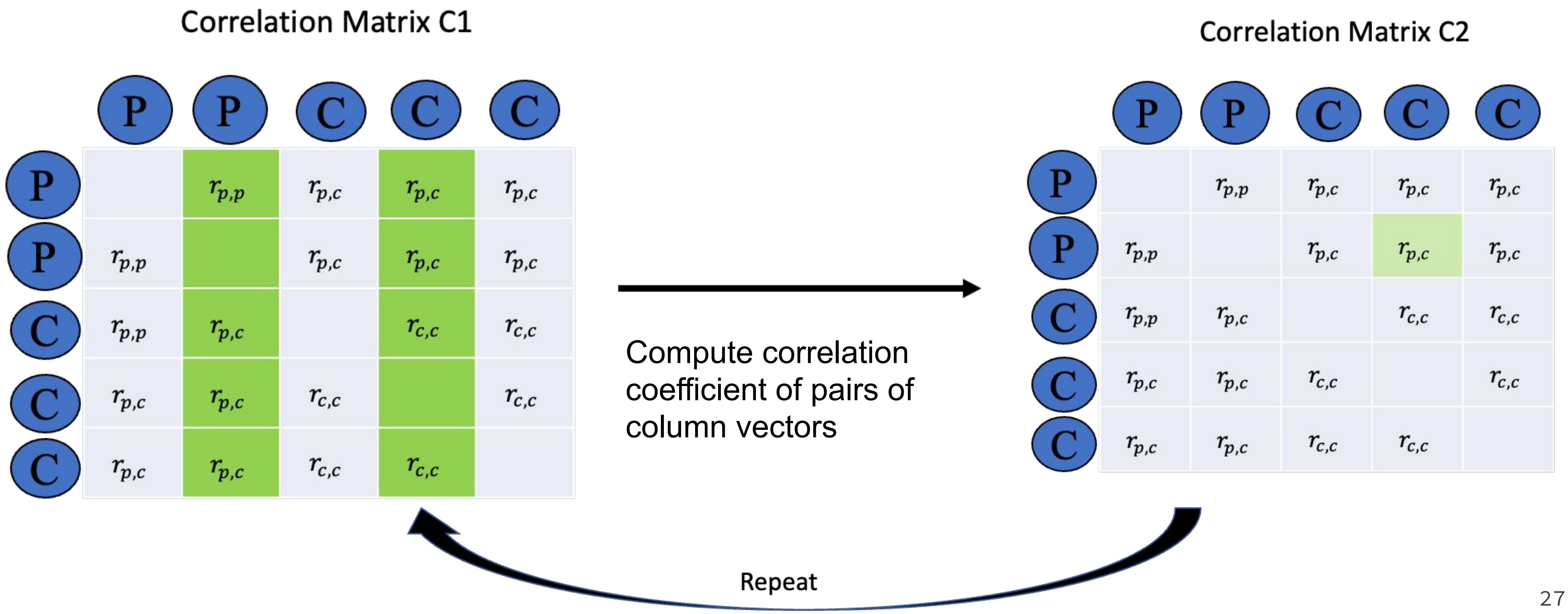
	P	P	C	C	C
P		$r_{p,p}$	$r_{p,c}$	$r_{p,c}$	$r_{p,c}$
P	$r_{p,p}$		$r_{p,c}$	$r_{p,c}$	$r_{p,c}$
C	$r_{p,p}$	$r_{p,c}$		$r_{c,c}$	$r_{c,c}$
C	$r_{p,c}$	$r_{p,c}$	$r_{c,c}$		$r_{c,c}$
C	$r_{p,c}$	$r_{p,c}$	$r_{c,c}$	$r_{c,c}$	

Compute correlation coefficient of pairs of column vectors

Correlation Matrix C2

	P	P	C	C	C
P		$r_{p,p}$	$r_{p,c}$	$r_{p,c}$	$r_{p,c}$
P	$r_{p,p}$		$r_{p,c}$	$r_{p,c}$	$r_{p,c}$
C	$r_{p,p}$	$r_{p,c}$		$r_{c,c}$	$r_{c,c}$
C	$r_{p,c}$	$r_{p,c}$	$r_{c,c}$		$r_{c,c}$
C	$r_{p,c}$	$r_{p,c}$	$r_{c,c}$	$r_{c,c}$	

Structural Equivalence



Structural Equivalence

Correlation Matrix C1

	P	P	C	C	C
P		$r_{p,p}$	$r_{p,c}$	$r_{p,c}$	$r_{p,c}$
P	$r_{p,p}$		$r_{p,c}$	$r_{p,c}$	$r_{p,c}$
C	$r_{p,p}$	$r_{p,c}$		$r_{c,c}$	$r_{c,c}$
C	$r_{p,c}$	$r_{p,c}$	$r_{c,c}$		$r_{c,c}$
C	$r_{p,c}$	$r_{p,c}$	$r_{c,c}$	$r_{c,c}$	

Repeated computation converges to +1s and -1s

Permute the rows and columns to get partitions

Converged Matrix

	P	P	C	C	C
P		+1	−1	−1	−1
P	+1		−1	−1	−1
C	−1	−1		+1	+1
C	−1	−1	+1		+1
C	−1	−1	+1	+1	

Partition the Nodes

	P	P	C	C	C
P		+1	−1	−1	−1
P	+1		−1	−1	−1
C	−1	−1		+1	+1
C	−1	−1	+1		+1
C	−1	−1	+1	+1	

After convergence, we always obtain two partitions

- +1 within the two partitions
- -1 between partitions

Partition the Nodes

	P	P	C	C	C
P		1.41	3.00	3.00	3.00
P	1.41		3.00	3.00	3.00
C	3.00	3.00		1.41	1.41
C	3.00	3.00	1.41		1.41
C	3.00	3.00	1.41	1.41	

Repeat correlation on partition

	P	P	C	C	C
P		+1	−1	−1	−1
P	+1		−1	−1	−1
C	−1	−1		+1	+1
C	−1	−1	+1		+1
C	−1	−1	+1	+1	

We can further obtain sub-partitions within each partition by repeating the iterative correlations:

- Euclidean distance within the partition
- Run CONCOR on Euclidean distance sub-matrix
- Permute rows and columns to get sub-partitions

These partitions are called blocks

Hence, the name blockmodeling

Data Reduction

Adjacency matrix → Density matrix → Image matrix

		$\mathscr{B}_1$	$\mathscr{B}_2$	$\mathscr{B}_3$	$\mathscr{B}_4$
		5 9 3 15 4 20	13 19 10 18	11 17 6 12 14 2 1 16 8	21 7
$\mathscr{B}_1$	5	- 0 0 0 0 1	1 1 1 1	1 1 1 0 1 1 1 1 1	1 1
	9	0 - 0 0 0 0	0 0 1 1	1 1 1 1 1 1 1 1 1	1 1
	3	0 1 - 0 1 1	0 0 1 1	1 1 1 1 1 1 1 0 1	1 1
	15	1 1 1 - 1 1	1 1 1 1	1 1 1 1 1 1 1 1 1	1 1
	4	0 0 0 0 - 1	0 0 1 1	1 1 1 1 0 1 1 1 1	1 0
	20	0 0 0 1 0 -	0 0 0 1	1 1 1 1 1 1 1 1 1	1 0
$\mathscr{B}_2$	13	1 1 0 0 0 0	- 0 0 1	0 0 0 0 1 1 1 0 0	0 0
	19	1 0 1 1 0 1	0 - 1 1	1 0 0 0 1 1 1 0 0	0 1
	10	1 0 1 1 1 1	1 1 - 1	1 1 0 0 0 1 1 1 1	0 0
	18	1 1 1 1 1 1	1 1 1 -	1 0 0 0 1 1 1 1 1	1 1
$\mathscr{B}_3$	11	0 0 0 0 0 0	0 0 0 0	- 0 0 0 0 1 1 0 0	0 1
	17	0 0 0 0 1 0	0 0 0 0	0 - 0 0 0 1 1 0 0	1 1
	6	0 0 0 0 0 0	0 0 0 0	0 0 - 0 0 0 0 0 0	1 0
	12	0 0 0 0 0 0	0 0 0 0	0 0 0 - 0 0 0 0 0	1 1
	14	0 0 0 0 0 0	0 0 0 1	0 0 0 0 - 1 0 0 0	1 1
	2	0 0 0 0 0 0	0 0 0 0	0 0 1 0 0 - 0 0 0	1 1
	1	0 0 0 0 1 0	0 0 0 1	0 0 0 0 0 1 - 1 1	1 0
	16	0 0 0 0 0 0	0 0 1 1	0 0 0 0 0 1 1 - 0	0 0
	8	0 0 0 0 1 0	0 0 1 1	1 0 1 0 0 1 0 0 -	1 1
$\mathscr{B}_4$	21	0 0 1 0 1 1	0 0 0 1	0 1 1 1 1 1 0 0 1	- 1
	7	0 0 0 0 0 0	0 0 0 1	1 1 1 1 1 1 0 0 0	1 -

	$\mathscr{B}_1$	$\mathscr{B}_2$	$\mathscr{B}_3$	$\mathscr{B}_4$
$\mathscr{B}_1$	0.367	0.625	0.944	0.833
$\mathscr{B}_2$	0.708	0.750	0.528	0.375
$\mathscr{B}_3$	0.056	0.167	0.194	0.722
$\mathscr{B}_4$	0.250	0.250	0.667	1.000

	$\mathscr{B}_1$	$\mathscr{B}_2$	$\mathscr{B}_3$	$\mathscr{B}_4$
$\mathscr{B}_1$	0	1	1	1
$\mathscr{B}_2$	1	1	1	0
$\mathscr{B}_3$	0	0	0	1
$\mathscr{B}_4$	0	0	1	1

Once partitions are obtained from CONCOR, rearrange the nodes by partition membership

Compute the edge density of the partition-to-partition relations (e.g., B1→B2 is 0.625)

Dichotomize the density matrix based on some criterion (e.g., alpha density criterion)

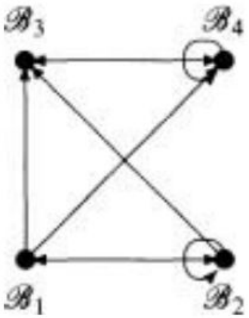
Wasserman and Faust (1994)

Block model

Image matrix

	$\mathscr{B}_1$	$\mathscr{B}_2$	$\mathscr{B}_3$	$\mathscr{B}_4$
$\mathscr{B}_1$	0	1	1	1
$\mathscr{B}_2$	1	1	1	0
$\mathscr{B}_3$	0	0	0	1
$\mathscr{B}_4$	0	0	1	1

Reduced graph



Reduced graph visualizes the connections among partitions

Each partition is a potential position of structural equivalence

The directed edges in the reduced graph represent roles

Useful for understanding the structure of the network

Blocks vs. Communities

Community detection and block models are similar in that they both summarize a network by grouping individuals

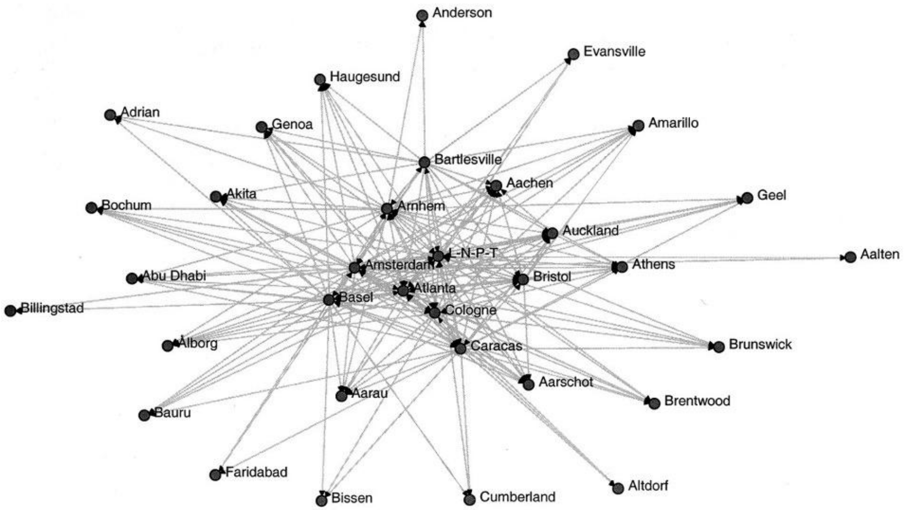
The main difference: Community detection tries to discover groups based on dense connections among members of a group

Block models are “agnostic” to within-block density and only care that block members exhibit similar connection patterns to other blocks.

Block Model Example: Global City Blocks

City-city ties based on headquarter-branch locations of the world’s 500 largest multinational firms

- Headquarter in Paris, branch in London: **London→Paris** tie
- Block model based on **regular equivalence**

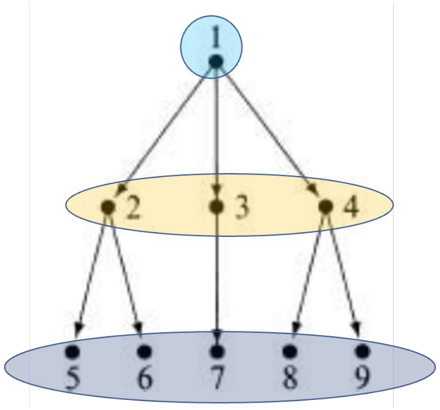


	Block Name	g_k	Out $_k$/Out	In $_k$/In	Self $_k$/Out $_k$	Position
1...	L-N-P-T	4	37.18	14.61	22.83	Primary
2...	Amsterdam	11	25.98	11.04	17.47	Primary
3...	Basel	27	20.49	6.87	15.15	Primary
4...	Atlanta	13	6.00	13.44	28.79	Primary
5...	Caracas	16	2.31	4.59	26.28	Primary
6...	Cologne	6	1.20	.95	12.10	Primary
7...	Bristol	2	.52	.47	.67	Primary
8...	Auckland	16	.30	8.65	53.49	High-status clique
9...	Athens	52	.04	8.64	27.27	High-status clique
10...	Bochum	12	.01	.55	75.00	High-status clique
11...	Arnhem	16	4.21	.37	4.85	Low-status clique
12...	Bartlesville	7	.54	.10	15.92	Low-status clique
13...	Aachen	79	.00	4.98	.00	Snob
14...	Brunswick	7	.00	.40	.00	Snob
15...	Evansville	4	.00	.12	.00	Snob
16...	Geel	2	.00	.07	.00	Snob
17...	Genoa	5	.00	.33	.00	Snob
18...	Aalten	818	.00	3.20	.00	Isolate

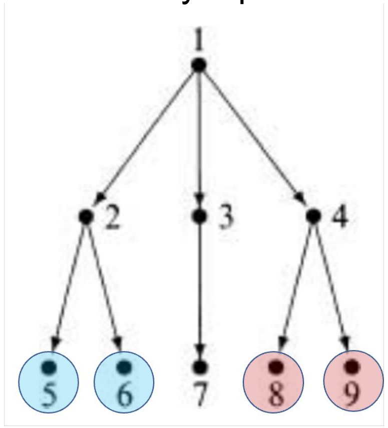
Alderson and Beckfield 2004

Regular Equivalence

regularly equivalent



structurally equivalent



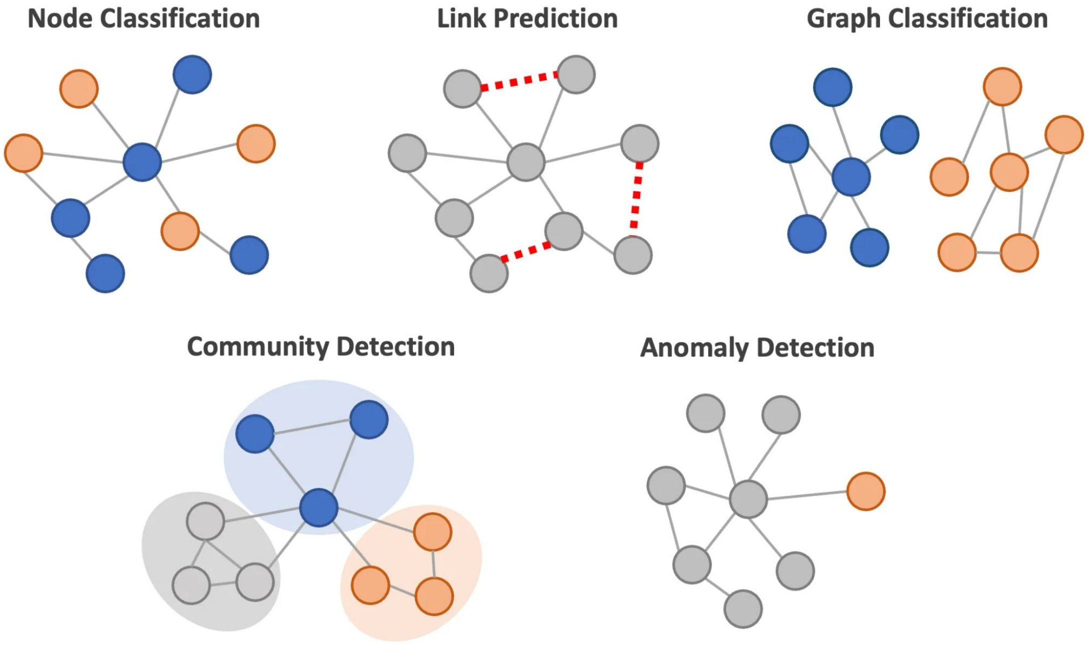
Regularly equivalent actors have identical ties to and from actors in other equivalence classes

Regular equivalence relaxes the stringent requirement of structural equivalence

Structural Equivalence in Machine Learning

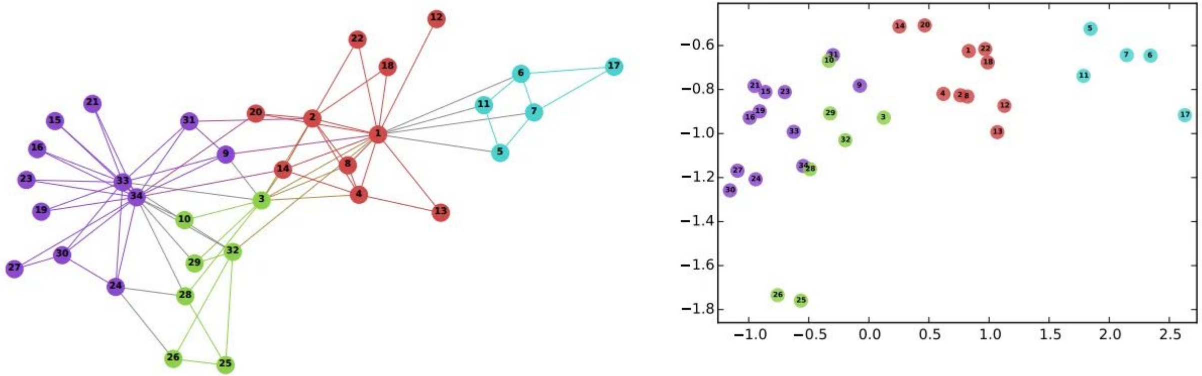
Structural Equivalence in Machine Learning

Graph representation learning: Graphs are used to extract useful information for prediction tasks at the node, edge, and graph levels



Structural Equivalence in Machine Learning

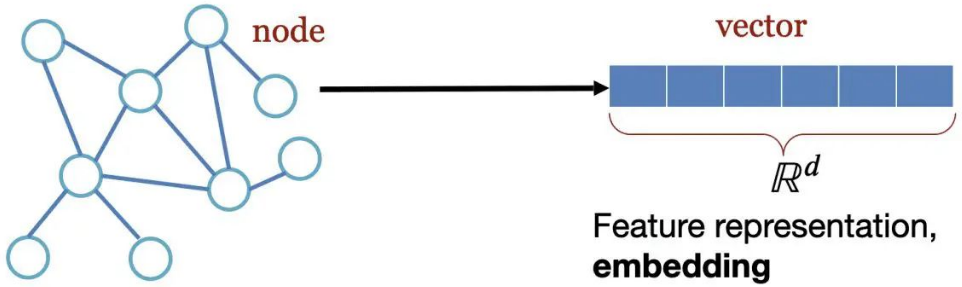
Idea: Take a network and somehow map the nodes in *k*-dimensional space



(a) Input: Karate Graph (b) Output: Representation

Structural Equivalence in Machine Learning

It is a process of reducing the graph information onto low-dimensional “features”
The features can be expressed as a vector of coordinates



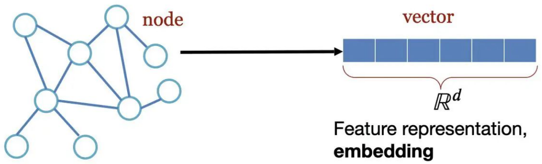
Structural Equivalence in Machine Learning

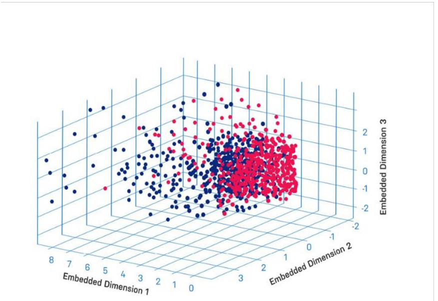
Each node in a network can be expressed in terms of a vector of length K (K dimensions)

The coordinates in this vector represents the position of a node

Nodes are embedded in this K-dimensional space

This embedding can be used for various prediction tasks





Node1	-0.3	0.1	0.244	0.156	-0.11	0.5
Node2	0.2	0.99	0.1	-0.003	0.314	-0.000001
Node3	-0.22	0.1	0.988	-0.142	0.339	-0.212
…	…	…	…	…	…	…

Structural Equivalence in Machine Learning

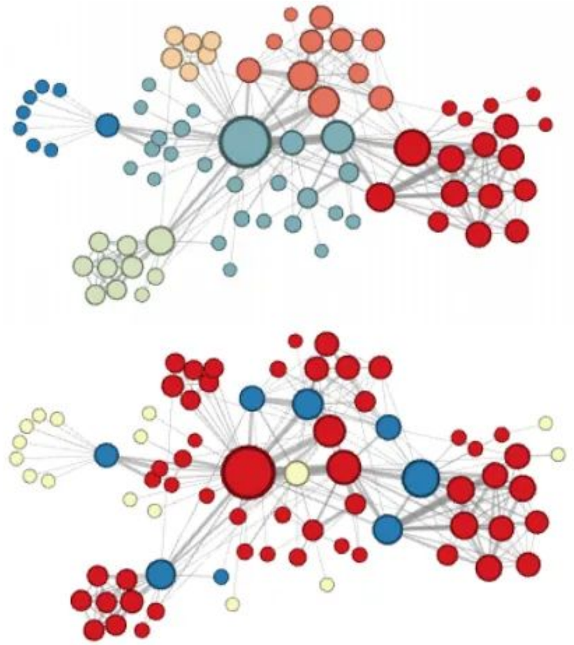


Figure 3: Complementary visualizations of Les Misérables coappearance network generated by *node2vec* with label colors reflecting homophily (top) and structural equivalence (bottom).

Node2Vec is a popular method for graph representation learning

Structural equivalence is an important consideration in how the algorithm works

A parameter can be tuned to prioritize homophily or structural equivalence in learning the node features

Grover and Leskovec 2016

Theoretical considerations

Dual nature of structural equivalence

People in the same position share similar experiences

- Structural equivalence means similar interaction patterns

Similarity can be helpful for learning in uncertain situations

But, it can also breed competition

- Structurally equivalent people are substitutable

Summary

Individuals can be grouped based on dense connectivity within vs. sparse connectivity between groups

An alternative way of thinking about groups is to partition the individuals based on the similarity of their ties to and from other individuals

The similarity in the connectivity links to the notion of social positions and roles

Block model is a useful method to hypothesize about social positions and role relations