

The background of the slide is a complex network diagram. It consists of numerous nodes (circles) of various sizes and colors (grey, white, yellow, green, blue, orange, purple, pink) connected by thin grey lines. Some nodes are highlighted with larger, colored circles around them, suggesting clusters or communities within the network. The overall structure is a dense, interconnected web of nodes and edges.

# Lecture 15 · Communities II, Social Networks I

## Networks, Crowds and Markets

# What is coming next

## Communities:

- Community detection algorithms
- The Stochastic Block Model (SBM)

## Social networks:

- Why social networks? Micro-mechanisms for tie formation.
- Strong vs. weak ties; triadic closure; bridges and local bridges.
- Strong Triadic Closure (STC): statement and consequence.
- Homophily: measurement, selection vs. influence.
- Affiliation (bipartite) networks and closures.

An agglomerative algorithm

# Ravasz agglomerative algorithm

Let  $B_i := \{j : d(i, j) \leq 1\}$ . Let  $A$  be the adjacency matrix.

We define node similarity using the *topological overlap*:

$$s_{ij} = \frac{|B_i \cap B_j|}{\min\{|B_i|, |B_j|\}} \in [0, 1], \quad (i \neq j).$$

- $s_{ij} = 0$  iff  $i, j$  are not connected and they share no neighbors.
- $s_{ij} = 1$  iff  $B_i \subseteq B_j$  or  $B_j \subseteq B_i$ .

# Ravasz agglomerative algorithm

Let  $B_i := \{j : d(i, j) \leq 1\}$ . Let  $A$  be the adjacency matrix.

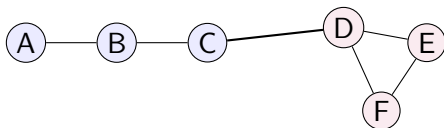
We define node similarity using the *topological overlap*:

$$s_{ij} = \frac{|B_i \cap B_j|}{\min\{|B_i|, |B_j|\}} \in [0, 1], \quad (i \neq j).$$

- $s_{ij} = 0$  iff  $i, j$  are not connected and they share no neighbors.
- $s_{ij} = 1$  iff  $B_i \subseteq B_j$  or  $B_j \subseteq B_i$ .

Many authors subtract  $A_{ij}$  in both the nominator and denominator to slightly down-weight the direct edge when  $i$  and  $j$  are linked ( $\tilde{s}_{ij}$ ).

## Toy example



| Pair         | $ B_i $ | $ B_j $ | $ B_i \cap B_j $ | $s_{ij}$ | $\tilde{s}_{ij}$ | $A_{ij}$ |
|--------------|---------|---------|------------------|----------|------------------|----------|
| A-B          | 2       | 3       | 2                | 1.00     | 1.00             | 1        |
| B-C          | 3       | 3       | 2                | 0.67     | 0.50             | 1        |
| C-D (bridge) | 3       | 4       | 2                | 0.67     | 0.50             | 1        |
| D-E          | 4       | 3       | 3                | 1.00     | 1.00             | 1        |
| D-F          | 4       | 3       | 3                | 1.00     | 1.00             | 1        |
| E-F          | 3       | 3       | 3                | 1.00     | 1.00             | 1        |
| A-C          | 2       | 3       | 1                | 0.50     | 0.50             | 0        |
| B-D          | 3       | 4       | 1                | 0.33     | 0.33             | 0        |
| C-E          | 3       | 3       | 1                | 0.33     | 0.33             | 0        |
| A-D          | 2       | 4       | 0                | 0.00     | 0.00             | 0        |
| A-E          | 2       | 3       | 0                | 0.00     | 0.00             | 0        |
| A-F          | 2       | 3       | 0                | 0.00     | 0.00             | 0        |

# Ravasz agglomerative algorithm: hierarchical clustering

We apply standard hierarchical clustering to  $S$ .

## Algorithm.

1. Compute the similarity matrix  $s_{ij} = \frac{|B_i \cap B_j|}{\min(|B_i|, |B_j|)}$  for  $i, j$ .
2. Treat each node as a separate cluster.

# Ravasz agglomerative algorithm: hierarchical clustering

We apply standard hierarchical clustering to  $S$ .

## Algorithm.

1. Compute the similarity matrix  $s_{ij} = \frac{|B_i \cap B_j|}{\min(|B_i|, |B_j|)}$  for  $i, j$ .
2. Treat each node as a separate cluster.
3. Find the two clusters with the largest average pairwise  $s_{ij}$ .
4. Merge them into a new cluster. Compute similarities between this new cluster and every other cluster using a chosen linkage rule:

$$s_{A,B} = \begin{cases} \max_{i \in A, j \in B} s_{ij}, & \text{(single linkage),} \\ \min_{i \in A, j \in B} s_{ij}, & \text{(complete linkage),} \\ \text{average}_{i \in A, j \in B} s_{ij}, & \text{(average linkage).} \end{cases}$$



# Ravasz agglomerative algorithm: hierarchical clustering

We apply standard hierarchical clustering to  $S$ .

## Algorithm.

1. Compute the similarity matrix  $s_{ij} = \frac{|B_i \cap B_j|}{\min(|B_i|, |B_j|)}$  for  $i, j$ .
2. Treat each node as a separate cluster.
3. Find the two clusters with the largest average pairwise  $s_{ij}$ .
4. Merge them into a new cluster. Compute similarities between this new cluster and every other cluster using a chosen linkage rule:

$$s_{A,B} = \begin{cases} \max_{i \in A, j \in B} s_{ij}, & \text{(single linkage),} \\ \min_{i \in A, j \in B} s_{ij}, & \text{(complete linkage),} \\ \text{average}_{i \in A, j \in B} s_{ij}, & \text{(average linkage).} \end{cases}$$

5. Repeat Step 3–4 until all nodes are merged into one cluster.

**Output.** The sequence of merges defines a *dendrogram*. Cutting it at a chosen similarity threshold yields the community partition.

## Average linkage: explicit calculations

**Step 1. Merge all cliques with all edges  $s = 1$ .**

These are:

$$\{A, B\}, \quad \{D, E, F\}.$$

Thus we obtain three initial clusters:

$$AB, \quad C, \quad DEF.$$

Cluster-cluster similarities (average linkage):

$$s_{AB,C} = \frac{1}{2}(s_{AC} + s_{BC}) = \frac{1}{2}\left(\frac{1}{2} + \frac{2}{3}\right) = \frac{7}{12} \approx 0.583,$$

$$s_{AB,DEF} = \frac{1}{6}(s_{A,D} + s_{A,E} + s_{A,F} + s_{B,D} + s_{B,E} + s_{B,F}) = \frac{1}{6} \cdot \frac{1}{3} = \frac{1}{18},$$

$$s_{C,DEF} = \frac{1}{3}(s_{C,D} + s_{C,E} + s_{C,F}) = \frac{1}{3}\left(\frac{2}{3} + \frac{1}{3} + \frac{1}{3}\right) = \frac{4}{9} \approx 0.444.$$

**Next merge:**  $AB$  and  $C$ , since  $s_{AB,C} \approx 0.583$  is the largest.

After merging  $AB$  and  $C$  the clusters are:  $ABC$  and  $DEF$ .

**Cluster similarity**  $s_{ABC,DEF}$ .

We average  $s_{ij}$  over all  $i \in \{A, B, C\}, j \in \{D, E, F\}$ . Non-zero terms are:

$$s_{B,D} = \frac{1}{3}, \quad s_{C,D} = \frac{2}{3}, \quad s_{C,E} = \frac{1}{3}, \quad s_{C,F} = \frac{1}{3}.$$

Hence

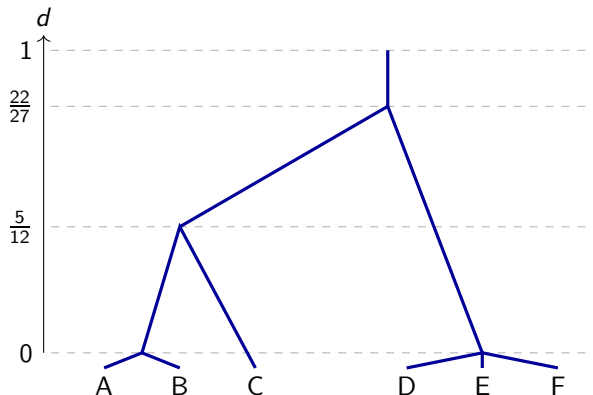
$$s_{ABC,DEF} = \frac{1}{9} \sum_{i \in ABC} \sum_{j \in DEF} s_{ij} = \frac{1}{9} \left( \frac{1}{3} + \frac{2}{3} + \frac{1}{3} + \frac{1}{3} \right) = \frac{5}{27} \approx 0.185.$$

**Final merge:** merge  $ABC$  with  $DEF$  at similarity  $s = 5/27$ .

# The dendrogram

**Dendrogram heights.** We use dissimilarity  $d = 1 - s$ :

$$d(AB, C) = 1 - \frac{7}{12} = \frac{5}{12} \approx 0.417, \quad d(ABC, DEF) = 1 - \frac{5}{27} = \frac{22}{27} \approx 0.815.$$



This dendrogram encodes a family of community structures (horizontal cuts). Choose the partition based on modularity.

# Modularity for the dendrogram partitions

**Modularity:** 
$$M = \frac{1}{2L} \sum_C \sum_{i,j \in C} \left( A_{ij} - \frac{k_i k_j}{2L} \right)$$

$$L = 6, \quad (k_A, \dots, k_F) = (1, 2, 2, 3, 2, 2).$$

Using this formula (summing only over pairs  $i, j$  in the same community):

- **Partition 1:**  $\{A, B\}, \{C\}, \{D, E, F\}$

$$M_1 = \frac{17}{72} \approx 0.24.$$

- **Partition 2:**  $\{A, B, C\}, \{D, E, F\}$

$$M_2 = \frac{23}{72} \approx 0.32.$$

- **Partition 3:** single community  $\{A, B, C, D, E, F\}$

$$M_3 = 0.$$

**Conclusion:** the best cut of the dendrogram (by modularity) is

$$\{A, B, C\}, \{D, E, F\}, \quad \text{since} \quad M_2 > M_1 > M_3.$$

A divisive algorithm

# Divisive community detection: Girvan–Newman algorithm

**Goal:** detect communities by removing *bridges* between them.

For each edge  $(i, j)$ , define its **edge betweenness**

$$b_{ij} = \sum_{s \neq t} \frac{\sigma_{st}(i, j)}{\sigma_{st}},$$

where  $\sigma_{st}$  is the number of shortest paths between nodes  $s$  and  $t$ , and  $\sigma_{st}(i, j)$  counts how many of them go through edge  $(i, j)$ .

- Edges with high  $b_{ij}$  tend to connect different communities.

# Divisive community detection: Girvan–Newman algorithm

**Goal:** detect communities by removing *bridges* between them.

For each edge  $(i, j)$ , define its **edge betweenness**

$$b_{ij} = \sum_{s \neq t} \frac{\sigma_{st}(i, j)}{\sigma_{st}},$$

where  $\sigma_{st}$  is the number of shortest paths between nodes  $s$  and  $t$ , and  $\sigma_{st}(i, j)$  counts how many of them go through edge  $(i, j)$ .

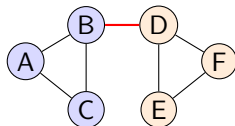
- Edges with high  $b_{ij}$  tend to connect different communities.

**Algorithm (top–down / divisive):**

1. Compute  $b_{ij}$  for all edges.
2. Remove the edge with the highest  $b_{ij}$ .
3. Recompute betweenness on the updated graph.
4. Repeat until the graph splits.



# Girvan–Newman: toy example



## Step 1. Compute edge betweenness

- Shortest paths inside each triangle use only local edges.
- All shortest paths between left and right triangles must go through  $(B, D)$ .
- $\Rightarrow$  edge  $(B, D)$  has the highest betweenness.

## Step 2. Remove $(B, D)$

Graph splits into two dense components:

$$\{A, B, C\}, \quad \{D, E, F\}.$$

**Communities recovered!**

## Girvan–Newman: selecting the best split

As in the Ravasz algorithm, the splits define a dendrogram.

The tree is constructed from top to bottom.

Cutting it defines a split.

Which split is best? Again, we can decide based on modularity.

As an exercise, you can go over this algorithm for the graph we considered for the Ravasz algorithm.

## Agglomerative vs. Divisive (summary)

|           | Agglomerative (Ravasz)      | Divisive (Girvan–Newman)            |
|-----------|-----------------------------|-------------------------------------|
| Direction | Bottom–up merges            | Top–down edge removals              |
| Input     | Node similarity $s_{ij}$    | Edge centrality $s_{ij}$            |
| Linkage   | Single / complete / average | Not applicable                      |
| Output    | Dendrogram of merges        | Dendrogram of splits                |
| Cost      | $O(N^2)$ typical            | $O(LN)$ – $O(N^3)$ (implementation) |

Both yield a dendrogram; choose the cut to get communities.

# Stochastic Block Model

# The Stochastic Block Model (SBM)

A simple generative model for networks with community structure.

## Definition ( Stochastic Block Model )

- $N$  nodes, each assigned to one of  $K$  groups:  $g_i \in \{1, \dots, K\}$ .
- Edge between  $i$  and  $j$  appears independently with probability

$$\Pr[(i, j) \in E] = p_{g_i, g_j}.$$

- The  $K \times K$  matrix  $P = (p_{ab})$  specifies connection probabilities between groups. Typically  $p_{kk} > p_{kl}$ .

# The Stochastic Block Model (SBM)

A simple generative model for networks with community structure.

## Definition ( Stochastic Block Model )

- $N$  nodes, each assigned to one of  $K$  groups:  $g_i \in \{1, \dots, K\}$ .
- Edge between  $i$  and  $j$  appears independently with probability

$$\Pr[(i, j) \in E] = p_{g_i, g_j}.$$

- The  $K \times K$  matrix  $P = (p_{ab})$  specifies connection probabilities between groups. Typically  $p_{kk} > p_{kl}$ .

## Applications:

- Benchmark for testing community detection algorithms.
- Statistical inference: given the observed network, estimate group labels  $\{g_i\}$  and/or  $P$ .

## SBM (2 groups): signal in the spectrum

**Model.** Two communities of sizes  $N_1, N_2$  ( $N = N_1 + N_2$ ).

$$\Pr[(i, j) \in E] = \begin{cases} p, & g_i = g_j, \\ q, & g_i \neq g_j. \end{cases} \quad \text{with } p > q.$$

Let  $A$  be the adjacency matrix (random) and  $B = \mathbb{E}[A]$  its expectation.

**Block structure of  $B$ :**

$$B = \begin{pmatrix} p\mathbf{1}_{N_1 \times N_1} & q\mathbf{1}_{N_1 \times N_2} \\ q\mathbf{1}_{N_2 \times N_1} & p\mathbf{1}_{N_2 \times N_2} \end{pmatrix},$$

a rank-2 matrix.

**Eigenvectors (intuition):**

- The *top eigenvector* points in a “degree/size” direction (it is close to  $\mathbf{1}$ ; exactly  $\mathbf{1}$  only if  $N_1 = N_2$ ).
- The *second eigenvector* is the **community indicator**:  $s_i = +1$  for group 1,  $s_i = -1$  for group 2. **Important insight.**

## Spectral clustering (basic idea)

With  $K \geq 2$  communities, their structure in the first  $K$  eigenvectors of  $B$ .

**Idea:** If we compute the leading eigenvectors of the *observed*  $A$ , they should align with those of  $B$  and reveal the partition.

This is the basic idea behind the **spectral clustering**.

Spectral clustering works because  $A$  concentrates around its block-structured mean and the “community” eigenvector survives the noise when separation is large enough.



# Social networks: basic concepts

# Motivation: what *creates* edges in social graphs?

- Edges (friendships, follows, coauthors) are *not* random: they arise from repeated social processes.
- Understanding these processes explains:
  - ▶ Information/job access, diffusion and influence, inequality of opportunity.
  - ▶ Community structure and clustering; small-world effects.
  - ▶ Predicting missing/future links (recommenders, growth).
- We will model edges via **micro-mechanisms** that we can test and use algorithmically.

# How do social ties form? (five recurring mechanisms)

1. **Triadic Closure:** friends of friends become friends.
2. **Focal Closure:** shared contexts (course, workplace) induce ties.
3. **Membership Closure:** people join contexts because friends are there.
4. **Homophily:** similar attributes  $\rightarrow$  higher linking probability.
5. **Social Influence:** after connecting, friends grow more alike.

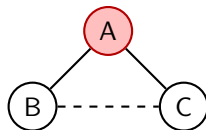
# Triadic Closure

## Definition

**Triadic Closure:** if two people have a common friend, the edge between them is more likely to form.

## Potential Reasons

- *Opportunity:* shared settings increase meetings.
- *Trust:* a mutual friend reduces risk.
- *Incentives:* social pressure to “close the triangle”.
- *Similarity:*  $B$ ,  $C$  may be close irrespective of their link to  $A$ .



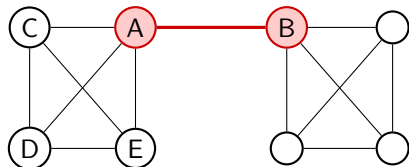
## Implications

- High clustering coefficient; redundant paths; robustness.
- Basis of many link-prediction features (common neighbors).

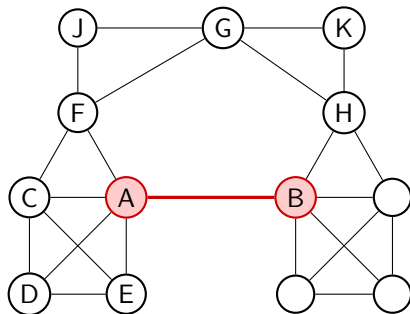
# Bridges and local bridges

**Bridge:** edge whose removal increases the number of connected components.

**Local bridge:** edge whose endpoints have no common neighbors.



Bridge



Local bridge

## Why they matter:

- Carry novel information across communities.
- Rare in dense social graphs (triadic closure tends to “fill” them).
- Often *weak ties* (acquaintances) act as local bridges (Granovetter).

# Strong vs. weak ties and Strong Triadic Closure (STC)

**Model:** label each edge as *strong* or *weak* (e.g. interaction frequency, emotional closeness).

**Strong Triadic Closure (STC):** A vertex satisfies STC if any neighbors with strong ties are connected by an edge (weak or strong).

**Consequence (Granovetter):** If  $A$  satisfies STC and has at least two strong ties, then any local bridge involving  $A$  must be a *weak* tie.

*Proof (by contradiction).* Suppose  $AB$  is strong tie and a bridge.  $A$  has at least one other strong tie  $AC$ . STC forces  $BC$  to exist but then  $AB$  cannot be a local bridge.  $\square$

# Homophily: measuring and interpreting

**Homophily:** people with similar attributes (age, interests, opinions, etc) connect at higher rates than random.

Homophily is one of the basic notions governing the structure of social networks.

# Homophily: measuring and interpreting

**Homophily:** people with similar attributes (age, interests, opinions, etc) connect at higher rates than random.

Homophily is one of the basic notions governing the structure of social networks.

**Quick test (binary attribute).** Let fraction  $p$  be group A,  $q = 1 - p$  group B. Under random mixing, fraction of cross-group edges  $\approx 2pq$ .

- If observed cross-group share  $\ll 2pq$ , evidence of homophily.
- Caveat: if degrees are very unequal, the  $2pq$  baseline is wrong; a better null model fixes the degree sequence (configuration model / assortativity).



# Homophily: measuring and interpreting

**Homophily:** people with similar attributes (age, interests, opinions, etc) connect at higher rates than random.

Homophily is one of the basic notions governing the structure of social networks.

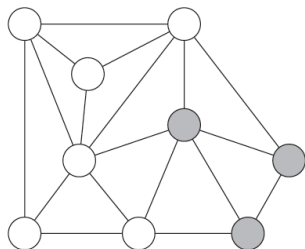
**Quick test (binary attribute).** Let fraction  $p$  be group A,  $q = 1 - p$  group B. Under random mixing, fraction of cross-group edges  $\approx 2pq$ .

- If observed cross-group share  $\ll 2pq$ , evidence of homophily.
- Caveat: if degrees are very unequal, the  $2pq$  baseline is wrong; a better null model fixes the degree sequence (configuration model / assortativity).

## Two explanations

- **Selection:** people prefer similar others  $\Rightarrow$  edges form due to similarity.
- **Influence:** similarity *increases* after the edge forms (behaviors diffuse).

## Mini example: homophily quick check



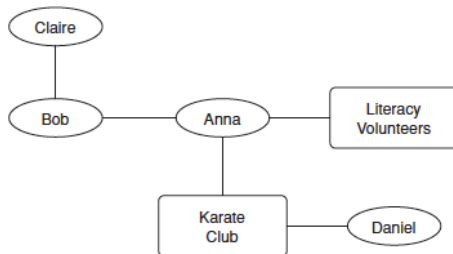
Suppose  $N = 9$  children: 6 girls ( $p = 2/3$ ), 3 boys ( $q = 1/3$ ).

Random mixing baseline for cross-gender edges:  $2pq = 4/9 \approx 0.444$ .

If the observed cross-gender share is, say,  $5/18 \approx 0.278$ , then it is well below  $2pq$   
 $\Rightarrow$  evidence of homophily.

# Affiliation networks and closures

An **affiliation network** is a bipartite graph between people and contexts (classes, clubs, workplaces).



**Triadic closure:** friends of a person tend to connect.

**Focal closure:** two people sharing a focus tend to connect.

**Membership closure:** a person joins a focus following a friend.

Positive and negative links

# Positive and Negative Relationships

In Social Networks, ties may represent:

- Friendship
- Collaboration
- Sharing of information
- ...

# Positive and Negative Relationships

In Social Networks, ties may represent:

- Friendship
- Collaboration
- Sharing of information
- ...

These relationships can either be:

- Positive relationships
- Negative relationships

The relationship in general have signed weights. We simplify here.

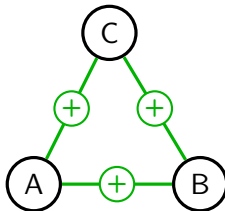
# Structural balance; signed triangles

We start by analyzing simple triangles with signed links.

## Structural balance; signed triangles

We start by analyzing simple triangles with signed links.

**Balanced**

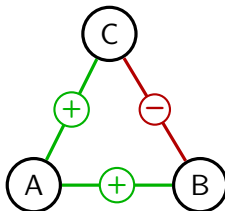




# Structural balance; signed triangles

We start by analyzing simple triangles with signed links.

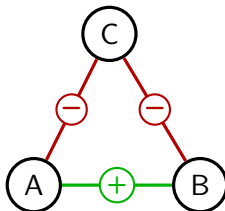
## Unbalanced



# Structural balance; signed triangles

We start by analyzing simple triangles with signed links.

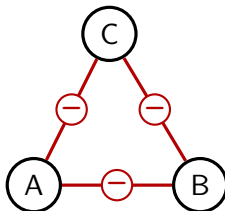
**Balanced**



# Structural balance; signed triangles

We start by analyzing simple triangles with signed links.

## Unbalanced



# Structural Balance Property and Polarization

Unbalanced triangles are sources of stress and psychological dissonance. Individuals strive to minimize them in their personal relationships.

A graph has **Structural Balance Property**, if every triangle is balanced.

**Balance Theorem:** If  $G$  is a complete graph with structural balance property then either all edges are positive, or, there exists a split of the nodes into two sets  $C_1/C_2$  such that the edges between  $C_1$  and  $C_2$  are negative and all the other edges are positive.

*Proof.* If all edges are positive, we are fine. Suppose that there is at least one negative edge. Take a vertex in  $G$  with a negative edge, call it  $a_1$ . Let  $C_2 = \{b_1, \dots, b_l\}$  be a set of nodes for which  $a_1 - b_i$  is negative. If  $l \geq 2$  consider the triangle  $\{a_1, b_i, b_j\}$  to show that  $b_i - b_j$  is positive for all  $i, j$ . Let  $C_1 = \{a_1, \dots, a_k\}$ . If  $k \geq 3$  consider the triangle  $\{a_1, a_i, a_j\}$  to show that  $a_i - a_j$  is positive for all  $i, j$ . Finally, consider the triangle  $\{a_1, a_i, b_j\}$  to show that  $a_i - b_j$  is negative for all  $i, j$ .

# Summary

- Social ties arise via **triadic**, **focal**, **membership** closures, and **homophily/influence**.
- **Strong vs. weak ties**: STC implies local bridges tend to be weak; weak ties carry novel information.
- **Affiliation** structure explains high clustering via shared foci.
- These mechanisms motivate **community structure** and power many **link-prediction** heuristics.