Network Science (Finals) - Jaish Khan

1. Network Projection

> **Network Projection** → Transforming a 2-mode network into a 1-mode network.

This transformation is done in network analysis because 2-mode networks are hard to analyze because they don't have as many analysis methods or network descriptors.

Characteristics and Implications

- **Bipartite Networks:** A two-mode network consists of two distinct, disjoint sets of nodes (U and V), where links exist only between a U-node and a V-node, not within the same set. Examples of two-mode networks include scientific collaboration networks (scientists and papers) or the Human Disease Network (diseases and genes).
- **Resulting Structure:** A significant consequence of network projection is that the resulting one-mode network often consists of **large fully-connected groups**. This occurs because all nodes connected to a single node in the two-mode structure become linked to each other in the one-mode projection.
- **Bias in Metrics:** This characteristic can introduce biases, especially affecting metrics like clustering coefficients, which rely on the existence of triangles (three connected nodes).

1.1. Methods of Network Projection

1.1.1. Binary Method

The simplest way to do network projection. You select one group of nodes (say, people) and draw a line (link) between any two people if they were both connected to the *same* item in the other group (say, attending the *same* event or writing the *same* paper). The final link is binary, meaning it's either **present (1)** or **absent (0)**, regardless of how many items they shared.

How to do projection via Binary Method?

1. Choose the set of nodes you want to project (e.g., people A, B, C).
2. For every pair of nodes (A and B, A and C, B and C), check if they share a connection to **at least one** node of the other type (e.g., an event or a paper).
3. If they share at least one, draw a link between them. The resulting network consists of only your chosen nodes and the new links.

1.1.2. Weighted Binary Method (Co-occurrence Counting)

Similar to the binary method, but instead of just drawing a line, you count how many items (co-occurrences) the two nodes shared. The resulting link is **weighted**, and that weight shows the strength of their connection.

How to do projection via Weighted Binary Method?

1. Choose the two nodes you are calculating the connection for (e.g., Author i and Author j).
2. Identify all the common items (e.g., papers) they both participated in.
3. Sum up the number of these common items. This sum is the weight (w_{ij}) of the new link between i and j.

Formula: The weight of the link (w_{ij}) is the sum ($\sum$) of occurrences (p) they share:

$$w_{ij} = \sum_{p} 1$$

Where p represents the nodes of the other kind (e.g., papers) that nodes i and j are connected to (their co-occurrences).

1.1.3. Newman Method

This method is commonly used for collaboration networks and recognizes that collaborating on a paper with only one other person creates a stronger bond than collaborating on a paper with *twenty* other people. It assigns a weight to the link based on shared items, but reduces the value of that tie according to the size of the original group (number of authors) that produced the item.

How to do projection by Newman Method?

1. For two nodes (i and j), look at one shared item (paper p).
2. Count the total number of participants (N_p) on that shared item.
3. The contribution of that single item to the link's weight is calculated as $1/(N_p - 1)$. If the item had 2 authors, the contribution is $1/(2-1) = 1$. If it had 5 authors, the contribution is $1/(5-1) = 0.25$.
4. Sum the contributions from all shared items to get the total link weight (w_{ij}).

Formula: The method for calculating collaboration weight is given as:

$$w_{ij} = \sum_{p} \frac{1}{N_p - 1}$$

Where N_p is the number of authors (nodes) on paper p.

1.1.4. Weighted Newman Method

A generalization of the original method, designed to handle cases where the links in the original 2-mode network already have weights (like number of messages, communication frequency). Similar to the original method, it incorporates reduction for the size of the group (N_p), but it also factors in the pre-existing weight of the interaction ($w_{i,p}$) from the two-mode structure.

Formula: The weights are calculated based on this approach by using the generalization given in the equation:

$$w_{ij} = \sum_p \frac{w_{i,p}}{N_p - 1}$$

Where $w{i,p}$ is the original weight attached to the link from node i to the shared resource p (e.g., weight of i's participation in event p)._

1.1.5. Sum Method

It is used when the original two-mode network had **weights** on its links already. When projecting, the new link between two people (i and j) is found by summing the specific flow/weight they directed toward the shared item. This method creates a directed, weighted, one-mode network, where the weight from A→B is not necessarily the same as the weight from B→A, shows difference in levels of interaction.

How to do projection by Sum method?

1. Choose two nodes, i and j.
2. Identify a shared common item (p).
3. Look up the original weight of the link from i to p (denoted $w_{i,p}$).
4. The new link weight w_{ij} is the sum of these weights across all shared intermediate nodes p.

Formula: The weighted two-mode network projection is formalized as:

$$w_{ij} = \sum_p w_{i,p}$$

Where $w{i,p}$ is the weight of the i-th node to the p-th event where i and j participated together.

Which projection method is the best

Choosing the right way to turn a two-mode network into a one-mode network is very important for good analysis.

Weighted projections are usually better than simple binary (yes/no) ones. But Why?

- Simple binary projection often creates huge fully-connected clumps, which distorts results (e.g., makes clustering look too high).
- Weighted methods (like Weighted Binary, Newman, or Sum) keep more of the original information.
- The **Newman method** is often the best choice because it reduces the strength of ties when many people share the same event or group — this fixes a lot of the bias.

2. Network Models

A network model is basically a recipe for how a network is formed or behaves. It tells you how nodes connect, why they connect, and what the resulting structure looks like.

They are essential tools in network science used to create a simple, mathematical pattern of something that cannot be seen directly. They allow researchers to show complex systems simply, derive network properties mathematically, and predict system behavior and outcomes.

> **Random Graph Theory**
>
> The whole **field** of mathematics that studies graphs where edges are placed randomly. It asks questions like: What does a typical random network look like? When do giant clusters appear? What is the average distance between nodes? How many triangles are there by chance?
>
> So “random graph theory” is the big umbrella subject. ER Random Network Model is a specific model.

2.1. ER Random Network Model

Erdős and Rényi Model, 1960

The oldest and simplest network model. It assumes that nodes connect randomly.

- **How it works** → You start with N nodes. For every possible pair of nodes, you flip a coin (with probability p) to decide if they connect.
  Alternatively, the network can be defined by having N nodes and a specific number of links (M), with M links randomly selected from all possible links.
- **What you get:**
  - Almost every dot ends up with roughly the same number of lines coming out of it
  - No “super-popular” stars or big hubs
  - Everyone is about equally connected
  - You can jump between any two dots in just a few steps (small world)
  - Friends of a friend are almost never friends with each other (very little clumping or triangles)
- **Real-life example** → Think of a group of strangers randomly bumping into each other and making connections.

> The structure of the resulting graph depends entirely on the value of p. The node degree distribution follows a **Binomial Distribution** which implies that the number

> of nodes with a specific degree rapidly decreases around the average degree, meaning all nodes have nearly the same degree. This is why you do not expect large hubs (highly connected nodes) in an ER random network. The network is also **homogeneously connected** with a short path length and a low clustering coefficient.

2. Small-World Model

Watts and Strogatz Model, 1998

A model created to capture two real-life properties that the ER model could not explain:
high clustering (friends of friends are usually friends) and short path lengths (anyone can be reached in a few steps).

- **How it works** $\longrightarrow$ Start with N nodes arranged in a ring, each connected to its nearest neighbors (like people only knowing their neighbors). Then with a probability p, rewire some links to random nodes, creating long-distance "shortcut" connections.
- **What you get:**
  - Nodes still form tight clusters of local friends.
  - But a few random shortcuts suddenly make the whole network feel very small.
  - You can reach distant nodes quickly even though you mostly connect locally.
  - The network becomes a blend of order + randomness.
- **Real-life example** $\longrightarrow$ You mostly know classmates, coworkers, neighbors (cluster), but you also know someone in another city or country (shortcut). This random long-distance friend makes the whole world feel "small."

> When $p = 0$, the network is a perfect ring: extremely high clustering but long path lengths. When $p = 1$, the network becomes similar to an ER random graph: low clustering, small path lengths. For intermediate p, the model produces the famous small-world effect: high clustering + short paths at the same time — something ER random networks cannot make.

3. Scale-Free Models

Real-world networks like the Internet do not follow random graph connections. Instead, they were found to be **scale-free**, where the node degree distribution follows a **power-law distribution** ($P(k) \sim k^{-\gamma}$). This distribution means that **few nodes (hubs) have many links**, while the majority of nodes have very small numbers of connections. Networks exhibiting this property show a **small average path-length** and a **high clustering coefficient**.

A. BA Scale-free Model

Barabási-Albert Model, 1999

The first simple model that explains why real networks (internet, social media, citations) contain a few **extremely popular hubs** instead of everyone being equally connected.

- **How it works** → The network **grows over time**: new nodes keep joining. Every new node prefers to attach to nodes that **already have many connections**. The rich get richer.
- **What you get:**
  - A few nodes become super-connected hubs.
  - Most nodes remain with only a few connections.
  - The degree distribution follows a power-law (a hallmark of scale-free networks).
  - The network naturally produces short path lengths.
- **Real-life example:** New websites tend to link to well-known pages (Google, Wikipedia). New social media users tend to follow celebrities first. As a result, popular nodes snowball into enormous hubs.

> It uses **Preferential Attachment** → The chance of connecting to a node is **proportional to its current degree**, leading to the "rich get richer" effect. However, BA only handles the **addition** of new nodes and does not consider node deletion, edge rewiring, or differences in node quality (fitness).

B. Fitness-Based Scale-free Model

Bianconi and Barabási

A refinement of the BA model that adds the concept of **node fitness**, explaining why some late-arriving nodes can still become highly connected hubs.

- **How it works** → Every node has a **fitness value** (η_i) representing how attractive, useful, or competitive it is. New nodes choose to connect based on **degree × fitness**. This means even a new node with high fitness can outcompete an old hub.
- **What you get:**
  - The network still becomes scale-free.
  - But some nodes become popular because of quality, not just age.
  - High-fitness nodes can become hubs very quickly.
- **Real-life example** → A brand-new YouTuber or TikToker goes viral overnight, not because they're old in the network, but because their content quality is high (fitness).

C. BBV Scale-free Model

A more advanced version of scale-free models that deals not only with **connections**, but also with **how strong** those connections are.

- **How it works** → Nodes attach preferentially to those with high **strength** (s_i), not just degree. Strength represents the **total weight** of a node's links (its traffic or load). When a new link arrives, existing weights in nearby nodes are redistributed and adjusted.
- **What you get:**
  - Nodes with heavy traffic ("busy nodes") keep getting busier → "busy get busier."
  - Both **degree distribution** and **weight distribution** become heavy-tailed.
  - Hubs form that are not only well-connected but also carry **large traffic loads**.
- **Real-life example** → Airports: Large airports have many routes (degree) and extremely heavy traffic (weight); new airlines prefer connecting to these big hubs.

Comparison of the Models

ER Random Model	Small-World Model	BA Scale-Free Model	Fitness-Based Model	BBV (Weighted Scale-Free)
Connections are made completely randomly	Mostly local neighbors, with a few shortcut links	New nodes link to popular nodes ("rich get richer")	New nodes link based on popularity × fitness	BA model but also grows connection weights
Nodes have similar degree (no big hubs)	High clustering + short paths	Creates big hubs and many small nodes	Late nodes with high fitness can still become big hubs	Hubs exist and their connections become stronger
Low clustering	Very high clustering	Low–medium clustering	Similar to BA	Higher weighted clustering
Node degrees follow a random pattern	Node degrees mostly regular	Power-law (some nodes extremely connected)	Power-law, shaped also by fitness	Power-law for degrees + weights
No "special" nodes	Neighborhood-based groups	Good for social media, web links	Good for viral success or sudden popularity	Good for airline networks, traffic, trade

3. Robustness and Vulnerability of Complex Networks

Complex systems show a remarkable ability to keep working even when individual components have failed. The main example of this is the Internet: hundreds of routers may malfunction simultaneously, yet global information flow remains stable.

This raises the fundamental question of **Where does this robustness come from, and how does network topology influence it?**

> **Real World Evidence**
>
> Research across various domains indicates that many real-world networks—including communication systems, biological processes, and social structures—are surprisingly resilient to random errors. Examples include:
>
> - **Technological:** Internet router networks and Autonomous System (AS) networks.
> - **Biological:** Cellular and metabolic networks.
> - **Social/Organizational:** Terrorist networks.

1.1. The Influence of Topology: Random vs. Scale-Free

The architecture of a network decides how its behaves under stress. The most important difference lies between **Random Networks** and **Scale-Free Networks**.

	Scale-Free Networks	Random Networks
Structure	Heterogeneous degree distribution: many low-degree nodes and a few highly connected hubs	Homogeneous degree distribution: nodes have similar degree and importance
Random Attack	**Robust** — random failures usually hit low-degree, non-critical nodes; network stays intact	**Moderately Vulnerable** — no special resilience like scale-free networks; random failures affect structure more uniformly
Targeted Attack	**Highly Vulnerable** — removing hubs rapidly breaks the network and reduces traffic capacity	**Robust** — since no node is uniquely critical, targeted removal does not cripple the network quickly
Nature	Dual nature: extremely robust to randomness but fragile to targeted attacks	Uniform: no special weakness, no special strength

> **Node removal** impacts robustness wayyy more than **Link Removal** because removing a node eliminates all its connected links.

1.2. Measuring Robustness

To quantify robustness, researchers simulate damage and measure how the network's structural properties change. This is useful for networks with weighted links and a relatively small number of nodes.

The General Procedure

1. **Simulation:** Remove a specific percentage of links (e.g., 5%, 10%, or 20%) randomly.
2. **Repetition:** Repeat the experiment multiple times (e.g., 20 runs) to ensure statistical reliability.
3. **Measurement:** Calculate centrality metrics before and after the removal to assess impact.

Key Metrics → **Degree**, **Strength**, **Closeness Centrality**, and **Betweenness Centrality**.

Weighted Betweenness Centrality

To understand the load on specific nodes, the following expression is used for weighted betweenness, where h represents the number of shortest paths:

$$C_B^{W^\alpha}(k) = \sum_{i \neq j} \frac{h_{ij}^{W^\alpha}(k)}{h_{ij}^{W^\alpha}}$$

Spearman Rank Correlation

To compare the "true" network (original state) against the "damaged" network. It compares the **ranking** of nodes based on centrality (e.g., Betweenness or Closeness) rather than raw values. High Correlation shows the network is structurally stable; the most important nodes remain important even after failures.

4. Link Prediction in Complex Networks

A fundamental challenge in network science. The goal is to estimate the likelihood of a connection forming between two nodes that are not currently connected.

Formally introduced by **Liben-Nowell and Kleinberg**, this problem focuses on understanding how networks evolve using "endogenous information": predicting future behavior based purely on the network's existing shape instead of external attributes.

> **Why it matters?**
>
> Since most real-world networks (social, biological, technological) are dynamic and constantly growing, link prediction is essential for:
>
> - **Forecasting Evolution:** Predicting which pairs of nodes (e.g., people, routers) will form links in the future.
> - **Recovering Missing Data:** Identifying connections that exist but are missing from the dataset due to incomplete sampling.
> - **Analyzing "Dark" Networks:** Uncovering hidden relationships in covert or criminal networks where data is intentionally obscured.
> - **Technological Optimization:** Predicting future data sharing or communication patterns between routers or servers.

4.1. Methods of Link Prediction

Most prediction methods rely on **structural similarity**, based on the fact that nodes with similar structural patterns are more likely to connect. These methods are generally divided into **Local** and **Global** indices.

4.1.1. Local Indices Measures

Local indices are computationally efficient because they rely only on the immediate neighborhood (connected peers) of the target nodes.

A. Common Neighbors (CN)

Used as a baseline benchmark for other methods, CN operates on the simple fact that two nodes sharing many friends are likely to become friends themselves. It counts the intersection of neighbors between node x and node y.

Formula: Where $\Gamma(x)$ represents the set of neighbors for node x

$$Score(x, y) = |\Gamma(x) \cap \Gamma(y)|$$

B. Jaccard Index

Used in information retrieval, the Jaccard Index refines the Common Neighbors approach by normalizing it. It measures the ratio of shared neighbors to total neighbors. It penalizes nodes that have a very high degree (too many connections), ensuring the similarity score isn't inflated just because a node is popular.

Formula:

$$J(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

C. Adamic–Adar Index (AA)

Made to measure similarity between personal homepages on the World Wide Web, this index refines the concept of common neighbors by weighting rare connections more heavily than common ones. A common neighbor that has very few other connections is considered more significant than a "hub" neighbor that connects to everyone.

Logic: It uses the inverse log frequency.

- If a common neighbor connects to only 2 people, the weight is high: $\frac{1}{\log(2)} \approx 1.4$.
- If a common neighbor connects to 5 people, the weight drops: $\frac{1}{\log(5)} \approx 0.62$.

4.1.2. Global Indices Measures

Global indices consider the entire topological structure of the network, not just immediate neighbors.

A. Katz Index

While Local indices look at direct neighbors (paths of length 2), the Katz index sums the ensemble of all paths connecting two nodes, exponentially penalizing longer paths.

5. Two Mode Clustering

Process of finding the tendency of nodes to form tightly connected groups in a **two-mode network**.

> **The Main Challenge in Two-Mode Clustering**
>
> In a two-mode network, nodes only connect **across** the two groups (like people → events), never **within** the same group. Because of this, a node's neighbors can't be connected to each other, so triangles can't form at all. Since normal clustering measures depend on triangles, you can't use standard (one-mode) local or global clustering coefficients on two-mode networks.

5.1. Approaches to Two-Mode Clustering

1. **Network Projection** → You convert a 2-mode network into a 1-mode network so you can use normal clustering measures. But this often creates big fully-connected groups, which messes up the results; especially anything that depends on triangles.
2. **Specially Designed Measures** → To avoid these problems, researchers created new clustering formulas made just for 2-mode networks so the results aren't biased.

> **Two-Mode Clustering Reinforcement, Robins & Alexander, 2004**
>
> It counts how many **4-cycles** exist compared to how many **3-paths** exist. A 4-cycle basically means two nodes of the same type took part in the **same two events**. But the problem is that this doesn't measure real **triadic closure** (closure among *three* nodes). It only shows how strongly **two nodes** are connected through events, not how a group forms. Also, it only works with **binary links** (no weights).

5.2. Global Two-Mode Clustering Coefficient

It fixes the above issue by measuring closure among **three nodes** in the main node set. It replaces the usual "triplets" with **4-paths** (open structures) and **closed 4-paths** (closed structures).

Weighted Two-Mode Clustering Coefficients
Opsahl also extended both global and local measures to handle **weights**.

1. **Weighted Two-Mode Global Coefficient** → Each 4-path gets a **value** based on its link weights. There are 5 methods:
    1. **Binary** → Pretends all connections are equal. Only cares if the links exist, not how strong they are.
    2. **AM (Arithmetic Mean)** → Takes the normal average of the two link weights. Treats “strong + weak” the same as “two medium”.
    3. **GM (Geometric Mean)** → Uses the “balanced” average (square root of product). Punishes pairs where one link is much weaker than the other.
    4. **Min (Minimum)** → Only counts the weaker of the two links. Very strict – the whole pair is only as good as its weakest part.
    5. **Max (Maximum)** → Only counts the stronger of the two links. Very generous – one strong tie makes the whole pair look good.
2. **Weighted Two-Mode Local Coefficient** → The local coefficient looks at **4-paths centered on one node**. The first and last nodes of the 4-path must be in the **same node set** as the focal node. This measure was also extended to use **weighted 4-path values**.

⑦ Which two-mode clustering co-efficient method is the most accurate?

The **Geometric Mean (GM)** method is considered the most accurate for two-mode clustering because it handles strong and weak link weights in a balanced, realistic way. It’s the standard used in most modern research and software.

6. Shortest Paths

The fundamental measurement of distance within a network. It is used as an underlying metric for centrality measures like betweenness and closeness.

> The **shortest path** between two nodes (i and j) is the path containing the fewest number of links. This path length is referred to as the distance, d_{ij}.

ⓘ Shortest Paths in Unweighted Networks

In an unweighted network, all links are treated equally. The distance between two nodes depends only on **how many steps** it takes to get from one to the other.

- If two nodes are directly connected → distance = **1**
- If you must pass through other nodes → distance = the **smallest number of steps** needed
  To find this shortest route, algorithms like **Breadth-First Search (BFS)** are used, because they naturally explore the network level by level and find the minimum number of steps.

⚠ Shortest Paths in Weighted Networks

Finding shortest paths is harder when links have **weights**. You can't just count how many links are in a path, because:

- A **longer path** with **strong links** might be better than
- A **shorter path** with **weak links**.

In weighted networks, a link with a **high weight** means a **strong connection**, so it should be treated as a **shorter** or "easier" step. To handle this, we use **Dijkstra's algorithm** (1959), which finds the path with the **least total cost**. But Dijkstra works only when **low cost = good link**, so we must **invert** the weights:

$$\text{cost} = \frac{1}{\text{weight}}$$

- Strong link (high weight) → **low cost**
- Weak link (low weight) → **high cost**

Then Dijkstra's algorithm finds the path whose **total inverted cost** is smallest. The weighted shortest-path distance between nodes (i) and (j) is:

$$d^W(i,j) = \min\left(\frac{1}{w_{i,h}} + \cdots + \frac{1}{w_{h,j}}\right)$$

Average Path Length → The average of the shortest paths between all pairs of nodes in the network.

Network Diameter → The LONGEST **shortest path** length found in the entire network. We basically find all the shortest paths and then the longest of those paths is the diameter.

ⓘ Difference between Binomial Distribution and Power Law Distribution

They describe two very different ways networks can be connected.

Binomial Distribution	Power-Law Distribution
Appears in ER random networks.	Appears in Scale-Free networks.
It is bell-shaped (starts and ends at the bottom and peaks in the middle)	It is long, heavy tailed (like a slope going down forever)
- Most have same degree - No Hubs	- Some have a very high degree - Hubs Exist
Rarely matches real networks	Fits many real systems
“Almost everyone is average”	“A few kings, millions of peasants”

THE END