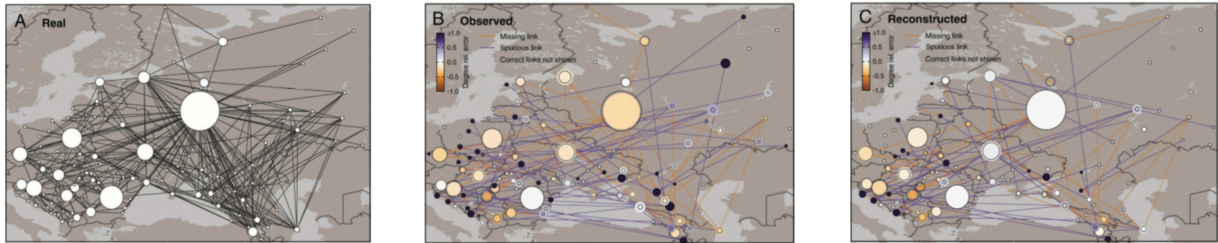


Link prediction by node similarity indices

Link prediction and *node similarity indices* aim to compute a score s_{ij} that is high for pairs of nodes i and j that are not connected but are likely to form a link in the future, and low for pairs of nodes that are unlikely to connect ([Kalyani et al., 2025](#)). In recommender systems (i.e. bipartite graph of users and items), this score is used to rank items so that for a given user i , the items j with the highest s_{ij} values are recommended.



In the following, we assume a simple undirected graph, although many indices can be generalized to other types of graphs.

Common neighbors index

Small-world networks are characterized by a high density of triangles, often explained by the principle of triadic closure in social networks. Consequently, two nodes are more likely to connect when they share many *common neighbors* ([Lorrain & White, 1971](#)).

$$s_{ij} = |\Gamma_i \cap \Gamma_j|$$

Jaccard coefficient index

The common neighbors index does not account for normalization. A standard way to compare two sets is through the *Jaccard coefficient* ([Jaccard, 1901](#)).

$$s_{ij} = \frac{|\Gamma_i \cap \Gamma_j|}{|\Gamma_i \cup \Gamma_j|} = \frac{|\Gamma_i \cap \Gamma_j|}{k_i + k_j - |\Gamma_i \cap \Gamma_j|}$$

Salton cosine similarity

Similarly, a common method to compare two vectors is by computing their dot product or *cosine similarity* ([Salton & McGill, 1983](#)).

$$s_{ij} = \cos \alpha_{ij} = \frac{A_i \cdot A_j}{\|A_i\| \|A_j\|} = \frac{|\Gamma_i \cap \Gamma_j|}{\sqrt{k_i k_j}},$$

where A_i is the i th row or column of A and α_{ij} is the angle between A_i and A_j .

Sørensen neighbors index

Salton cosine similarity normalizes the number of common neighbors by the geometric mean of the degrees. Alternatively, the normalization can be done using the arithmetic mean instead ([Sørensen, 1948](#)).

$$s_{ij} = \frac{|\Gamma_i \cap \Gamma_j|}{\frac{1}{2}(k_i + k_j)} \propto \frac{|\Gamma_i \cap \Gamma_j|}{k_i + k_j}$$

Leicht similarity index

Leicht similarity index normalizes the number of common neighbors by the expected number in a corresponding random graph ([Leicht et al., 2006](#)).

$$s_{ij} = \frac{|\Gamma_i \cap \Gamma_j|}{\frac{k_i k_j}{n-1}} \propto \frac{|\Gamma_i \cap \Gamma_j|}{k_i k_j},$$

where $\frac{k_i k_j}{n-1}$ is the probability of a link between nodes i and j in the configuration model.

Resource allocation index

The *resource allocation* index accounts for the fact that nodes are more likely to share a high-degree neighbor ([Zhou et al., 2009](#)).

$$s_{ij} = \sum_{x \in \Gamma_i \cap \Gamma_j} \frac{1}{k_x}$$

Adamic-Adar index

Adamic-Adar index further accounts for the fact that some nodes in scale-free networks have very high degrees ([Adamic & Adar, 2003](#)).

$$s_{ij} = \sum_{x \in \Gamma_i \cap \Gamma_j} \frac{1}{\log k_x}$$

Preferential attachment index

The scale-free networks with a power-law degree distribution $p_k \sim k^{-\gamma}$ are believed to result from *preferential attachment*, where nodes tend to connect to high-degree nodes called hubs ([Barabasi & Albert, 1999](#)).

$$s_{ij} = k_i k_j$$

Random walk index

Node similarity can also be measured using *random walks with restarts* or personalized PageRank ([Tong et al., 2006](#)).

$$s_{ij} = p_i^{(j)} + p_j^{(i)}$$

Community index

Many real networks are organized into *communities* of densely connected nodes, with relatively few links between communities. Consequently, links are more likely to form within communities than between them ([Yan & Gregory, 2011](#)).

$$s_{ij} = 1 \text{ if } c_i = c_j \quad \& \quad s_{ij} = 0 \text{ if } c_i \neq c_j$$

Block model index

The *block model* index extends this concept to arbitrary clusters of nodes, such as communities, bipartite structures, or core–periphery organizations. In particular, the probability of a link is determined by the density of connections within and between clusters ([Doreian et al., 2010](#)).

$$s_{ij} = \frac{m_{c_i}}{\binom{n_{c_i}}{2}} \text{ if } c_i = c_j \quad \& \quad s_{ij} = \frac{m_{c_i c_j}}{n_{c_i} n_{c_j}} \text{ if } c_i \neq c_j,$$

where n_{c_i} and m_{c_i} are the number of nodes and edges in cluster c_i , and $m_{c_i c_j}$ is the number of edges between clusters c_i and c_j .