Measures and Metrics, Nodes

Measures and Metrics

Given a chosen *metrics* to simplify and represent a studied network, we could count on our innate ability to find patterns from a visualisation of the network to discover some facts of the network by inspecting it.

However, this approach does not scale the larger the network gets.

A better approach is to define *mathematical measures* that capture interesting features of network structure quantitatively, boiling down large volumes of complex structural data into numbers that are an indication of the studied phenomena.

Kinds of Metrics • Binary Scale

The simplest and most popular kind of metrics. Conventionally, 1 indicates the presence of a relationship and 0 indicates its absence.

Being the “ground floor" of the information, it can always be obtained starting from another metric, defining a threshold value (cut-off point) below which all values are reported to 0 and above to 1.

The information that is lost in this way is often compensated by the greater ease of analysis.

Kinds of Metrics • Multi-category Nominal Scales

This metric indicates for each relation the “type” that it assumes, with respect to multiple-choice list (example: lover, friend, colleague, enemy, ...).

The analysis can be carried out at the level of a single type (e.g., networks that have “lover” as a link between the nodes), with effects on the measures (e.g., reduction of density) of which it is important to be aware.

Kinds of Metrics • Ordinal Scales

The simplest ordinal metric refers to a three-value scale, of the type “-1 0 +1”, where:

- - 1 implies the presence of a “negative” relationship (e.g., “aversion of one actor to another”);
- 0 indicates indifference;
- +1 implies the complementary situation to the negative one.

Other ordinal measures refer to larger scales, e.g., the Likert one or based on the request to each actor to express the order with which (s)he would like to have relations with the other nodes of the network.

Ordinals can always be brought back to one of the previous scales, losing information.

Kinds of Metrics • Scalar Scales

Scalar metrics are useful when handling values representing either physical quantities - like metres, kilograms, seconds, amperes, moles - or information units and units of account - money, goods, services, assets, labor, income, expenses.

Scalar measures have been developed more recently, through the adaptation of algorithms originally created for the much simpler binary metrics.

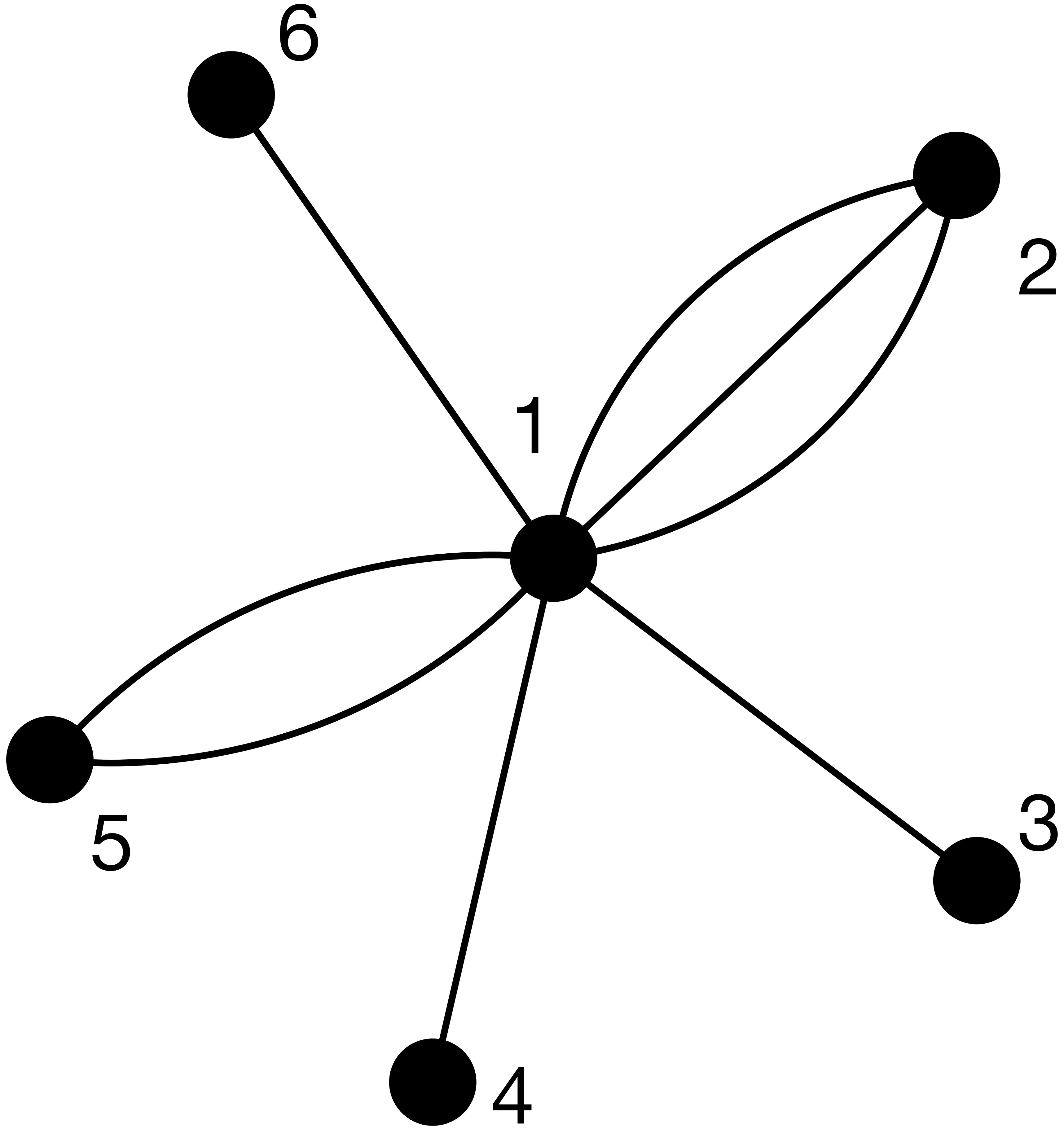
Degree

One of the simplest measures is the degree of a node.

In an undirected network, the degree of a node is the **number of edges connected** to it.

E.g., in a social network of friendships between individuals a person's degree is the number of friends they have.

Despite its simplicity, the degree is one of most useful and most widely used of network concepts and it plays an important role in other measures.



$$deg(1) = 8 \ \cdots \ deg(5) = 2 \ \cdots \ deg(3) = 1$$

$$deg(i) = \sum_{j=1}^{n} A_{ij}$$

Centrality

Centrality measures answer to the question:

“Which are the most important or central nodes in a network?”

Of course, there are many possible definitions of “importance” and there are correspondingly many centrality measures for networks.

Centrality • Degree Centrality

One of the simplest centrality measure for a node in a network is just its **degree**.

In **directed networks**, nodes have both an **in-degree** and an **out-degree**, and both may be useful as measures of centrality in the appropriate circumstances.

Although degree centrality is a simple centrality measure, it can be very illuminating.

For example, in a social network those individuals who have many followers might have more influence, more access to information, or more prestige than those who have fewer.

A non-social network example is the use of citation counts in the evaluation of scientific papers. The number of citations a paper receives from other papers, which is its in-degree in the directed citation network, gives a quantitative measure of how influential the paper is.

Centrality • Eigenvector Centrality

In many circumstances a node's importance in a network is increased by having connections to other nodes that *are themselves important*.

For example, you might have only one friend in the world, but if that friend is the president of the United States then you yourself may be an important person. Thus centrality is not only about how many people you know but also who you know.

Eigenvector centrality is an extension of degree centrality that takes this factor into account. Instead of just awarding one point for every network neighbour a node has, eigenvector centrality awards a number of points ***proportional to the centrality scores of the neighbours***.

Centrality • Eigenvector Centrality

Considering an undirected network of *n* nodes, the eigenvector centrality x_i of node *i* is proportional to the sum of the eigenvectors centralities of *i*'s neighbours.

Mathematically

$$x_i = \sum_{j \in neighbours(i)} x_j$$

Since it is a sum, a node can achieve a high eigenvector centrality either by having a lot of neighbours with modest centrality or a few neighbours with high centrality (and everything in between) - the intuitive interpretation of this is that nodes can be influential either by reaching a lot of nodes or by reaching just a few, highly-influential nodes.

Centrality • Eigenvector Centrality

Mathematically

$$x_i = \alpha \sum_{j \in neighbours(i)} x_j$$

proportionality factor

Centrality • Eigenvector Centrality

Mathematically

$$x_i = \alpha \sum_{j \in neighbours(i)} x_j \quad \Rightarrow \quad x_i = \alpha \sum_{j=1}^{n} A_{ij} \, x_j$$

proportionality
factor

Centrality • Eigenvector Centrality

Mathematically

$$x_i = \alpha \sum_{j \in neighbours(i)} x_j \quad \Rightarrow \quad x_i = \alpha \sum_{j=1}^{n} A_{ij} \, x_j$$

proportionality factor

Centrality • Eigenvector Centrality

Mathematically

We need to solve a system of linear equations, which leads us to the matrix notation, for all values $x_1, \cdots, x_n$

$$x_i = \alpha \sum_{j \in neighbours(i)} x_j \quad \Rightarrow \quad x_i = \alpha \sum_{j=1}^{n} A_{ij} \, x_j \quad \Rightarrow \quad \alpha^{-1} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

proportionality factor

Centrality • Eigenvector Centrality

Mathematically

We need to solve a system of linear equations, which leads us to the matrix notation, for all values $x_1, \cdots, x_n$

$$x_i = \alpha \sum_{j \in neighbours(i)} x_j \quad \Rightarrow \quad x_i = \alpha \sum_{j=1}^{n} A_{ij} \, x_j \quad \Rightarrow \quad \alpha^{-1} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

proportionality factor

Still, we do not know what values $x_1, \cdots, x_n$ assume.

Centrality • Eigenvector Centrality

Mathematically

$$x_i = \alpha \sum_{j \in neighbours(i)} x_j \quad \Rightarrow \quad x_i = \alpha \sum_{j=1}^{n} A_{ij} \, x_j \quad \Rightarrow \quad \alpha^{-1} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

proportionality factor

We need to solve a system of linear equations, which leads us to the matrix notation, for all values $x_1, \cdots, x_n$

Still, we do not know what values $x_1, \cdots, x_n$ assume. However, our last transformation let us understand that the vector of centralities is one of the possible eigenvectors of the matrix A

$$\underset{\text{Eigenvalue}}{\kappa} \, \overset{\text{Eigenvector}}{\mathbf{x}} = \underset{\text{Matrix}}{A} \, \overset{\text{Eigenvector}}{\mathbf{x}}$$

Centrality • Eigenvector Centrality

Mathematically

$$x_i = \alpha \sum_{j \in neighbours(i)} x_j \Rightarrow x_i = \alpha \sum_{j=1}^{n} A_{ij} x_j \Rightarrow \alpha^{-1} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

proportionality factor

We need to solve a system of linear equations, which leads us to the matrix notation, for all values $x_1, \cdots, x_n$

Still, we do not know what values $x_1, \cdots, x_n$ assume. However, our last transformation let us understand that the vector of centralities is one of the possible eigenvectors of the matrix A

$$\kappa \mathbf{x} = A \mathbf{x}$$

Eigenvector

Eigenvector

Eigenvalue

Matrix

$$\kappa \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Centrality • Eigenvector Centrality

Mathematically

$$x_i = \underset{\text{proportionality factor}}{\alpha} \sum_{j \in neighbours(i)} x_j \quad \Rightarrow \quad x_i = \alpha \sum_{j=1}^{n} A_{ij}\, x_j \quad \Rightarrow \quad \alpha^{-1} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

We need to solve a system of linear equations, which leads us to the matrix notation, for all values $x_1, \cdots, x_n$

Still, we do not know what values $x_1, \cdots, x_n$ assume. However, our last transformation let us understand that the vector of centralities is one of the possible eigenvectors of the matrix A

$$\underset{\text{Eigenvalue}}{\kappa}\,\overset{\text{Eigenvector}}{\mathbf{x}} = \underset{\text{Matrix}}{A}\,\overset{\text{Eigenvector}}{\mathbf{x}}$$

$$\kappa \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \kappa^{-1} A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Now, we “just” need to find what values $\mathbf{x}$ and κ assume.

Centrality • Eigenvector Centrality

How do we choose κ and $\mathbf{x}$?

Assuming we want our centrality values to be **all positive**, then we can use the Perron–Frobenius theorem, by which

> for a square matrix with all elements non-negative (like our adjacency matrix A) there exists a unique largest eigenvalue (κ) and the corresponding eigenvector ($\mathbf{x}$), called leading, that have strictly positive components

The eigenvector centrality x_i of node i is the i^{th}element of the leading eigenvector of the adjacency matrix and the value of the constant κ is the leading eigenvalue.

Centrality • Eigenvector Centrality

Although we fixed $\mathbf{x}$ and κ, the centrality measure remains arbitrary within a multiplicative constant.

This is not a problem, when we use that measure within the nework. Indeed, the **multiplicative constant does not matter** much, as we are applying transformations to the values in our adjacency matrix that maintain their proportions.

However, when using eigenvector centrality in absolute terms (e.g., when comparing different matrices) we need to normalise those values, to make them comparable. One possibility, here, is to *normalise the centralities*, e.g., by requiring that they sum to n (which ensures that the average centrality stays constant as the network gets larger).

Centrality • Eigenvector Centrality

For the case of **directed networks**, the eigenvector centrality poses some complications due to the **asymmetricity** of adjacency matrices. This translates into two sets of eigenvectors, left and right, and two leading eigenvectors.

Which to choose among the two depends on the reason of the calculation of the centrality measure. The **right eigenvector** measures centrality as **bestowed by others** to the node. The **left eigenvectors** measures centrality as **connections of the node to the others**.

For example, in the Web and in citation networks, a good indication of the importance of a node is how many nodes point to it. However, if we consider transport networks, hubs that connect to a lot of locations tend to be more important.

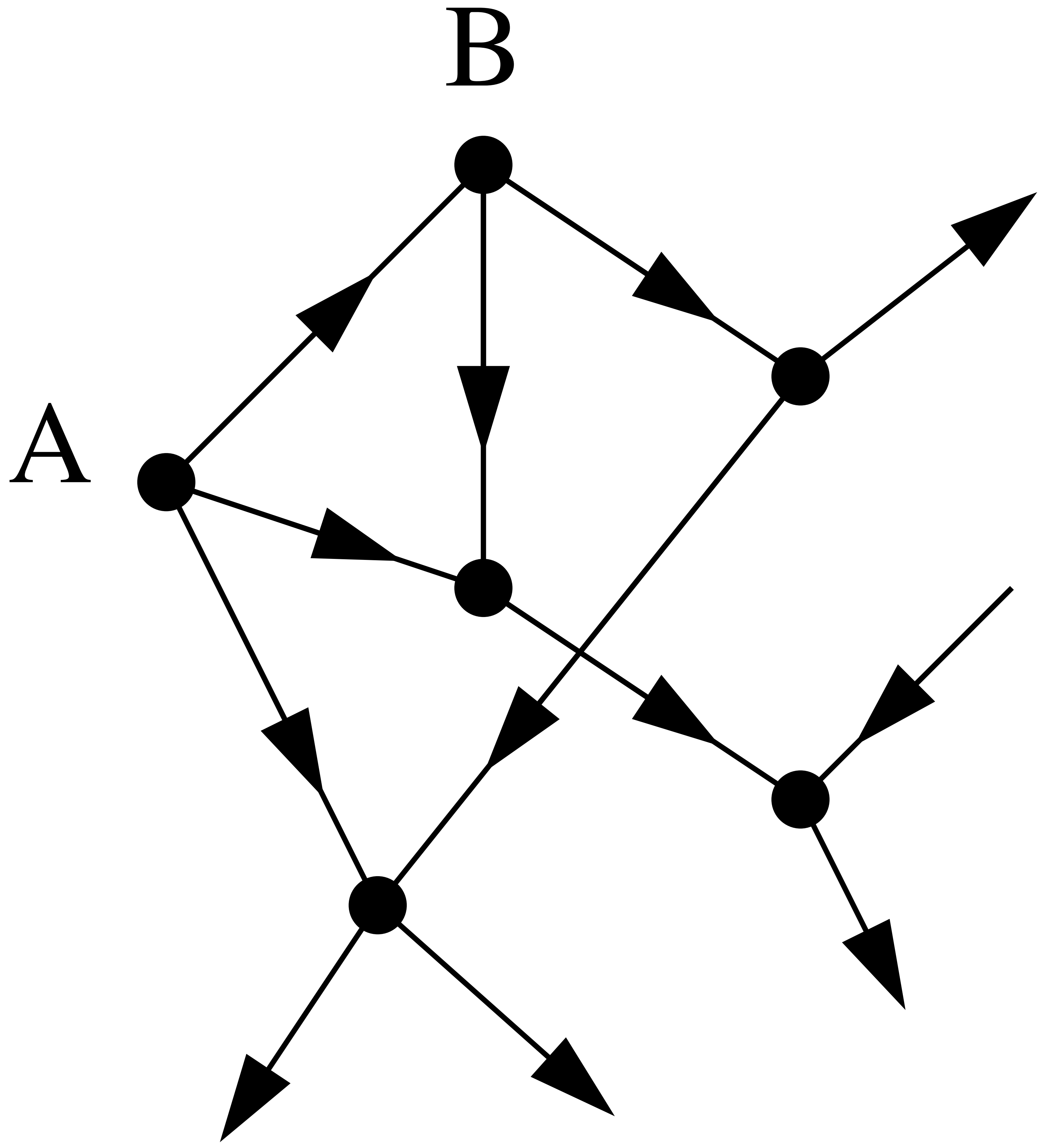
Centrality • Problems of Eigenvector Centrality

There are still problems with this definition of centrality.

Let us illustrate them with an example.

Consider the left (inbound) eigenvector centrality on the network on the right. Since Node A has only outgoing edges and no ingoing ones, its eigenvector centrality is zero. Node B, which has one ingoing edge, also have eigenvector centrality zero, because calculated from its only ingoing edge from A, which has centrality zero.

This simple example points to a problem: a node may be pointed by others that themselves are pointed by many more, but if the trail ends have in-degree zero, the final value of the centrality will be zero.



Centrality • Katz Centrality (aka PageRank)

To solve the problem of zero-trailing in eigenvector centrality for directed networks, Katz proposed a centrality measure that gives each node a small amount of centrality "for free" regardless of its position in the network or the centrality of its neighbours

Mathematically

$$x_i = \alpha \sum_j A_{ij}\ x_j + \beta$$

In the formula, α is related to the eigenvalue but β is the "for free" part that all nodes receive. By adding β, we ensure that even nodes with zero in-degree still get the non-zero centrality β, which they can "pass" to the other nodes they point to. Thus, any node that is pointed by many others has a high centrality, even if it is not in a strongly connected component.

Centrality • Katz Centrality (aka PageRank)

To solve the problem of zero-trailing in eigenvector centrality for directed networks, Katz proposed a centrality measure that gives each node a small amount of centrality "for free" regardless of its position in the network or the centrality of its neighbours

Mathematically

$$x_i = \alpha \sum_j A_{ij} \; x_j \boxed{+ \beta}$$

The caveat here is that, while in eigenvector centrality the multiplicative constant for κ did not matter, now α (which contains κ) conflicts with β. Indeed, if $\alpha \rightarrow 0$ all nodes have the same centrality β. Past κ_1^{-1}, with κ_1 being the largest eigenvalue of A, the centrality diverge. Thus $0 < \alpha < 1/\kappa_1$.

Centrality • Katz Centrality (aka PageRank)

While α shall be contained within 0 and κ_1^{-1}, there is no broad agreement on which value α should take.

Interestingly, Katz centrality captures the degree and eigenvector centralities at its extremes: the former for $\alpha \to 0$, the latter for $\alpha \to \kappa_1^{-1}$.

Concretely, this means that small values of α favour strongly connected components while values closer to κ_1^{-1} give small non-zero values to nodes that are not in strongly connected components of size two or more.

Centrality • Katz Centrality (aka PageRank)

One problem with Katz centrality is that, **if a node with high Katz centrality** has edges pointing to many others, then **all of those others also get high centrality**. Concretely, a high-centrality node pointing to one million others gives all one million of them high centrality.

To see the practice of this issue, consider websites like *Amazon* or *eBay which* link to the web pages of thousands of manufacturers and sellers. Now, following Katz centrality, if Amazon is an important website and has a link to a semi-unknown website, also that website receives a high Katz centrality.

Would that be a good representation of the reality of centrality in the Web?

Centrality • Katz Centrality (aka PageRank)

To solve this problem, we define a variant of the Katz centrality where we derive the centrality of the neighbours as proportional to their centrality *divided by their out-degree*. Then nodes that point to many others pass only a small amount of centrality on to each of those others, even if their own centrality is high.

In mathematical terms, this centrality is defined as

$$x_i = \alpha \sum_j A_{ij} \frac{x_j}{od(j)} + \beta$$

With the caveat of defining the out-degree $od(\cdot)$ to assign 1 to nodes whose out-degree is 0

Centrality • Katz Centrality (aka PageRank)

This centrality measure is commonly known as *PageRank*, named after Larry Page, co-founder of Google.

Google uses PageRank to estimate the importance of web pages, which the search engine lists by “importance” (centrality).

The added ingredient of dividing by the out-degrees of pages ensures that pages that simply point to an enormous number of others do not pass much centrality on to any of them.

Centrality • Katz Centrality (aka PageRank)

One might wonder if part of Google's "secret sauce" is their multiplier α Katz centrality.

Google has been pretty transparent on this, stating that its search engine uses a value of α equal to 0.85.

It is not clear that there is any rigorous theory behind this choice.

More likely, it is just a shrewd guess based on experimentation to find out what works.

Centrality • Hubs and Authorities

The centrality measures for directed networks seen so far all follow the same basic principle: high centrality goes with *being pointed* by others (with high centrality).

In some cases, nodes are highly central when they *point to other highly central* ones.

In this kind of networks there are two types of “important” nodes:

- *authorities:* nodes that hold useful resources;

- *hubs:* nodes that are gateways toward the most resourceful authorities.

Authorities may also be a hubs (and vice versa).

Centrality • Hubs and Authorities

Hyperlink-induced topic search or *HITS* is a centrality measure that gives each node *i* in a directed network two different centrality scores: the *authority centrality* x_i and the *hub centrality* y_i, defined using the constants α and β and by swapping the indices of the matrix element (since the hub centrality of a node *i* is defined by the nodes it points to)

$$x_i = \alpha \sum_j A_{ij}\ y_i \qquad\qquad y_i = \beta \sum_j A_{ji}\ x_j$$

Interestingly, hub- and authority-centrality circumvent the problems that ordinary eigenvector centrality has with directed networks: in hub-and-authority approach nodes not pointed by any others have authority centrality zero but they can still have non-zero hub centrality and the nodes that *they* point to can then have non-zero authority centrality by virtue of being pointed.

Centrality • Closeness Centrality

Differently from the previous centrality measures, based on nodes' degree, closeness centrality uses the shortest paths in networks, measuring the mean distance from a node to other nodes.

Let us first define the mean distance of a node *i*.

Suppose d_{ij} is the shortest distance from node *i* to node *j*. Then the mean shortest distance from *i* to every node in the network is

$$\ell_i = \frac{1}{n} \sum_j d_{ij}$$

Centrality • Closeness Centrality

Thus, the mean distance ℓ_i is not a centrality measure per-se, since it **gives low values to more central nodes and high values for less central ones**. To be used as a centrality, we can use the inverse of ℓ_i rather than ℓ_i itself.

This inverse is called the ***closeness centrality*** C_i:

$$C_i = \ell_i^{-1} = \frac{n}{\sum_j d_{ij}}$$

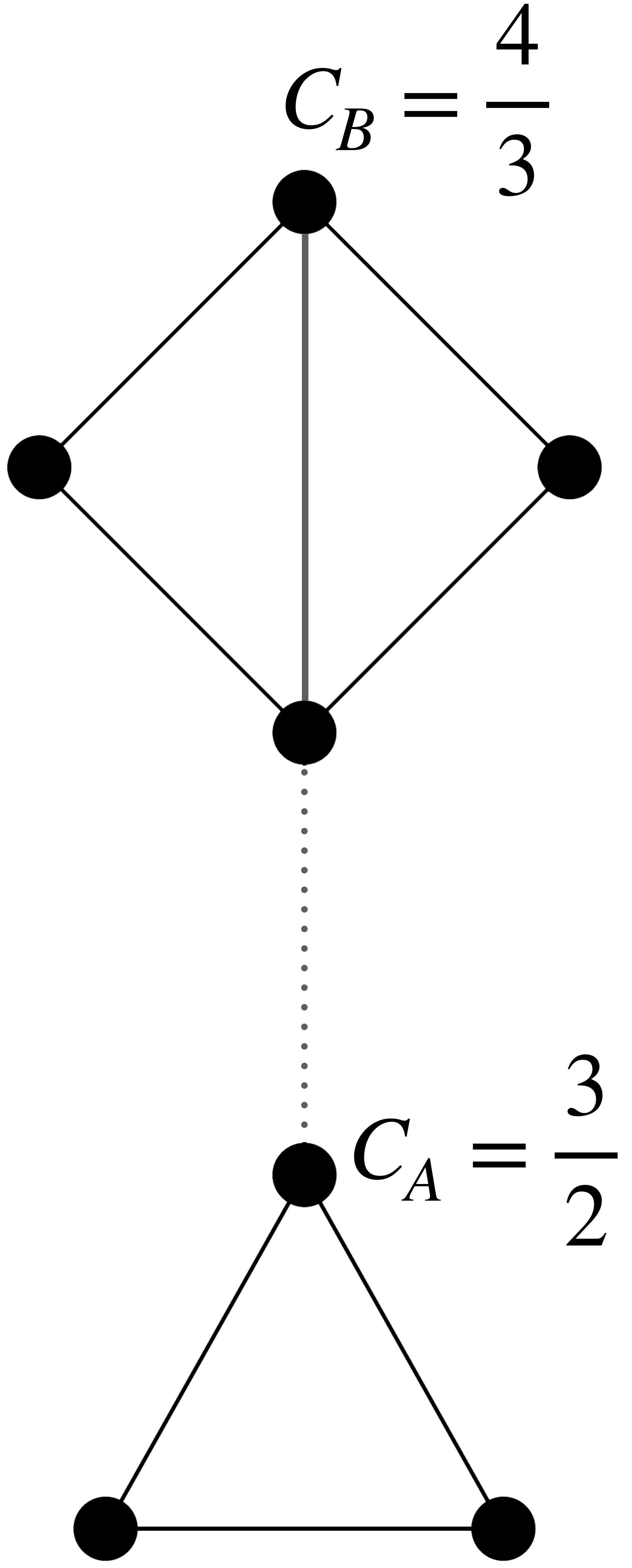
For closeness centrality, the smaller ℓ_i is, the better, i.e., ℓ_i takes small values for nodes that are separated from others by only a short distance on average — the assumption is that small-mean-distance nodes might have more direct influence on others or better access.

Centrality • Closeness Centrality

Closeness centrality has a problem with networks with more than one component and non-existent paths set to infinite. There, when nodes belong in different components ℓ_i is infinite and C_i is zero.

To solve this problem, it is possible to **average over only those nodes in the same component** as *i* (and *n* indicates the number of nodes in the component).

This gives a finite measure, but one that has its own problems. E.g., distances tend to be smaller between nodes in small components. Mathematically, those small-component nodes get lower values of ℓ_i and higher closeness centrality than their counterparts in larger components. On the contrary, nodes in small components are usually assumed to be *less* well connected than those in larger ones and should therefore be given lower centrality.



Centrality • Closeness Centrality

An alternative solution is to redefine closeness in terms of the ***harmonic mean distance*** (*the reciprocal of the arithmetic mean of the reciprocal*) between nodes, i.e., the average of the inverse distances:

$$\ell'_i = \frac{n-1}{\sum_{j(\neq i)} \frac{1}{d_{ij}}} \qquad C'_i = \frac{1}{n-1} \sum_{j(\neq i)} \frac{1}{d_{ij}}$$

Where we exclude from the sum the term for *j* =? *i (and thus count $n-1$ nodes)* to avoid to get an infinite division.

The measure, when $d_{ij} = \infty$ because *i* and *j* are in different components, zeroes the term and drops it; moreover it gives more weight to nodes that are close to *i* than to those far away.

Centrality • Betweenness Centrality

Betweenness centrality, also based on shortest paths, measures the extent to which a node lies on paths between other nodes. The assumption here is that paths lying on “trafficked” shortest paths have a more central role in the network, as gateways favoured by their closeness to (reach) the other nodes.

Mathematically, betweenness centrality of undirected networks can be expressed as follows.

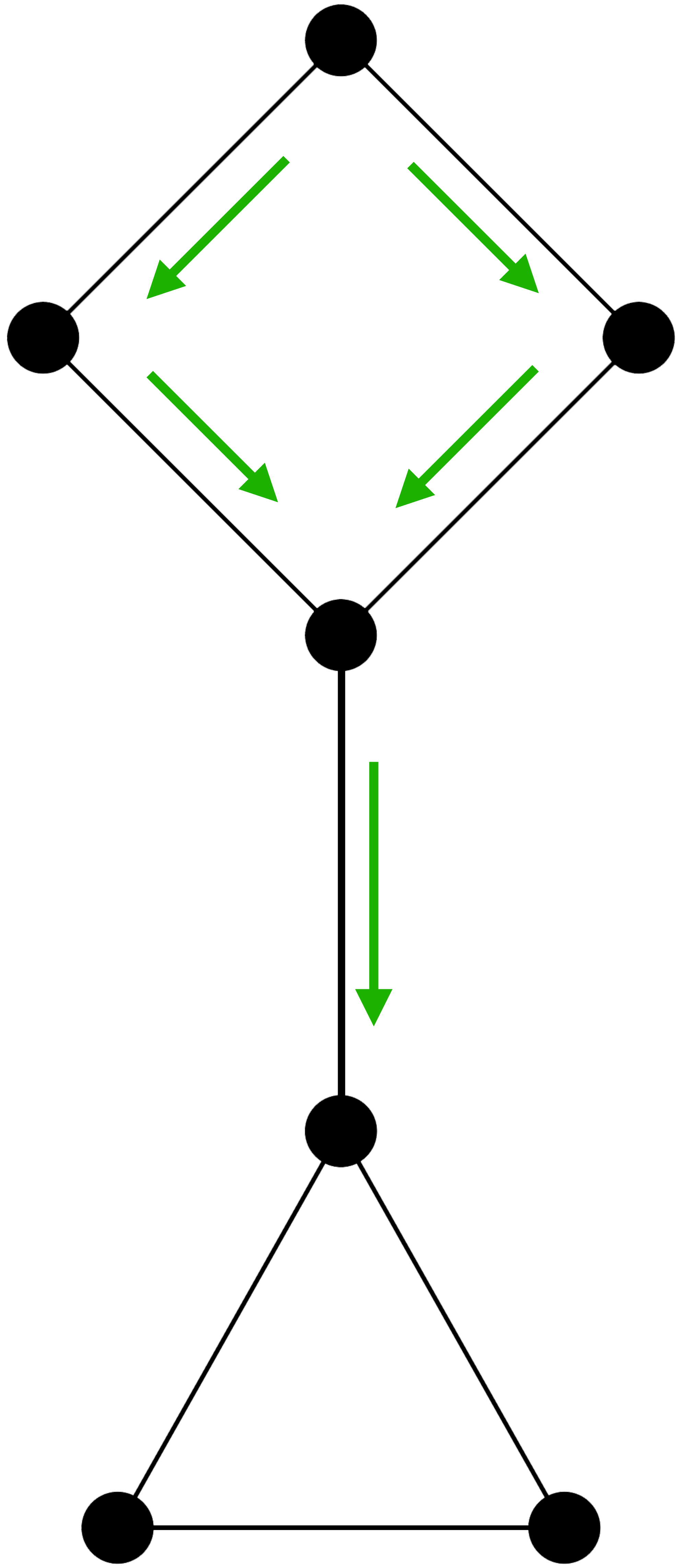
Suppose that we have an undirected network in which there is at most one shortest path between any pair of nodes and let n^i_{sd} be 1 if node *i* lies on the shortest path from the source *s* to the destination *d* and 0 if it does not or if there is no such path.

The betweenness centrality x_i is given by the formula:

$$x_i = \sum_{sd} n^i_{sd}$$

For all shortest paths

Centrality • Betweenness Centrality

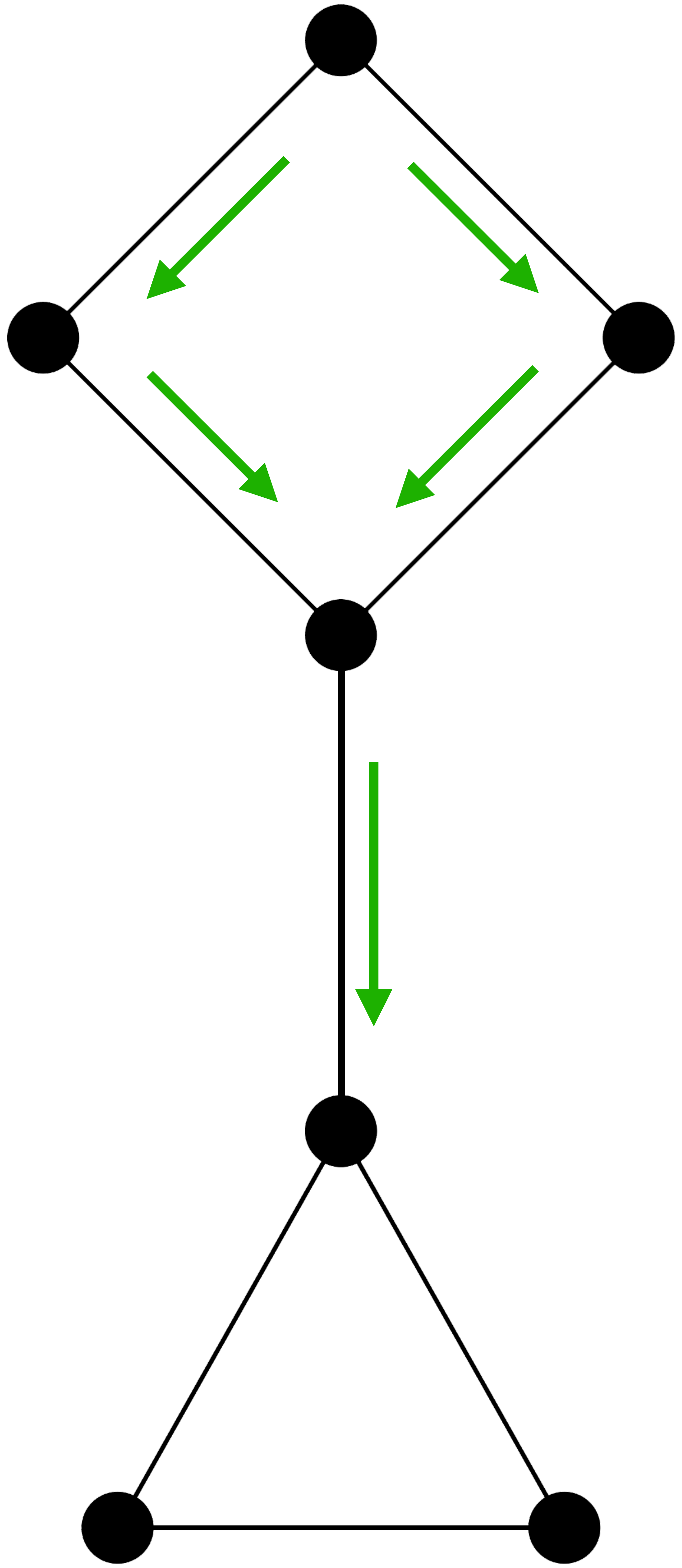


Since it is possible for two shortest paths between the same pair of nodes to overlap, we refine n^i_{sd} to be the number of shortest paths from *s* to *d* that pass through *i* and define g_{sd} as the total number of shortest paths from s to d, obtaining

$$x_i = \sum_{sd} \frac{n^i_{sd}}{g_{sd}}$$

assuming as convention $n^i_{sd}/g_{sd} = 0$ if both n^i_{sd} and g_{sd} are zero, the newly-defined value of x_i corresponds to the average rate of the “traffic” that passes through node *i.*

Centrality • Betweenness Centrality



The values of betweenness considered so far are raw numbers of paths, but it is sometimes convenient to normalise betweenness. One natural choice is to normalise the path count by dividing it by the total number of (ordered) node pairs, which is n^2, so that *betweenness becomes the fraction* (rather than the number) *of paths that run through a given node*:

$$x_i = \frac{1}{n^2} \sum_{sd} \frac{n^i_{sd}}{g_{sd}}$$

The refined measure, besides normalising betweenness, has the additional benefit of limiting the values of centrality between 0 and 1.

Groups of Nodes

Many networks divide naturally into groups or communities:

- networks of people divide into groups of friends, co-workers, or business partners;
- the World Wide Web divides into groups of related web pages;
- biochemical networks divide into functional modules.

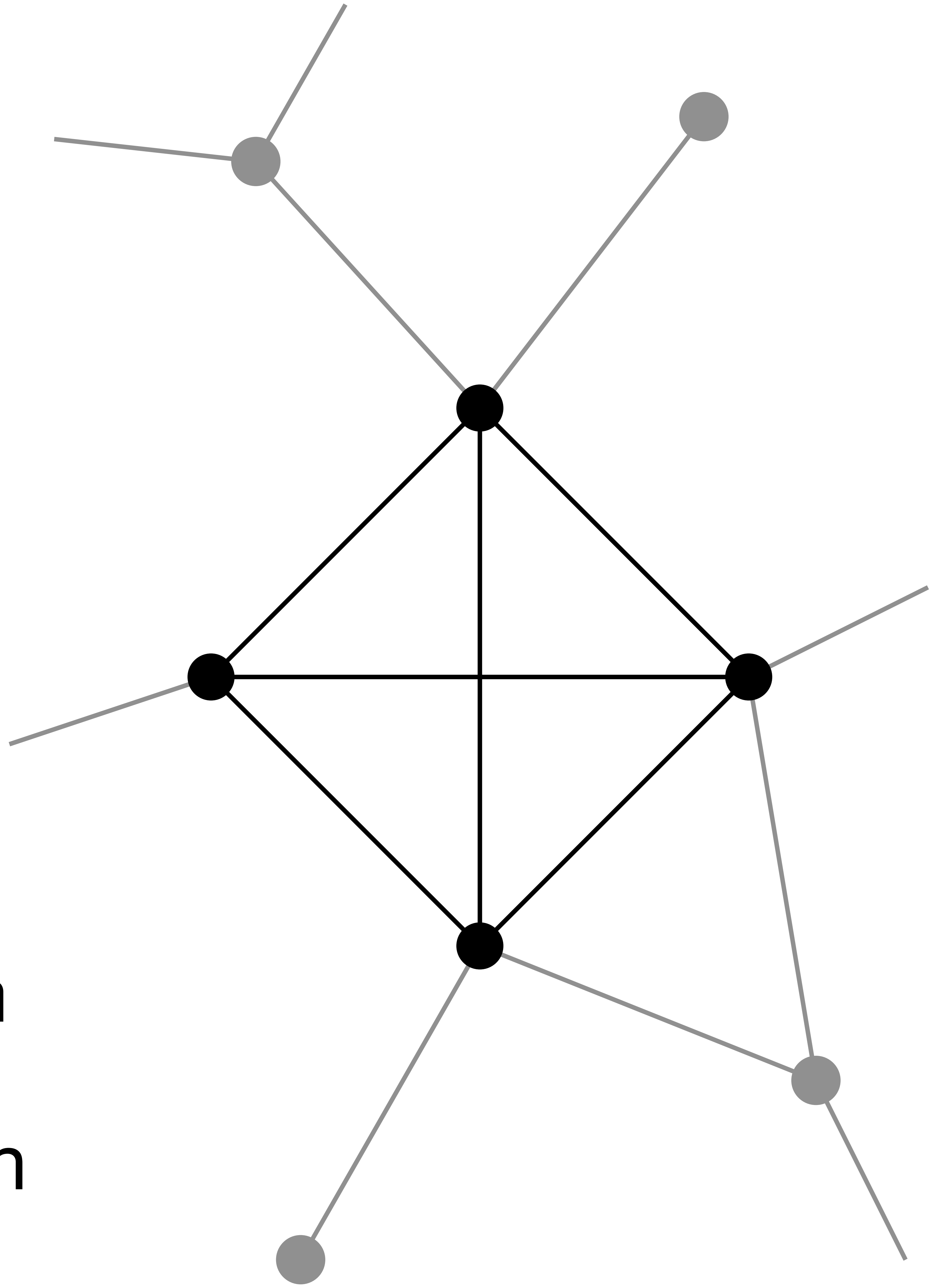
Besides calculating their centrality, it is possible to apply measures to nodes to detect their membership to one or more constituent groups.

Groups of Nodes • Cliques

A *clique* is a set of nodes within an undirected network such that **every member of the set is connected by an edge to every other**. Cliques can overlap, meaning that they can share one or more of the same nodes.

The occurrence of a clique in an otherwise sparsely connected network is normally an indication of a highly cohesive subgroup — like the members of a family or a set of co-workers in an office.

It is also possible that many circles of acquaintances form only *near-cliques*, rather than perfect cliques. There may be some members of a group who are unacquainted, even if most members know one another.



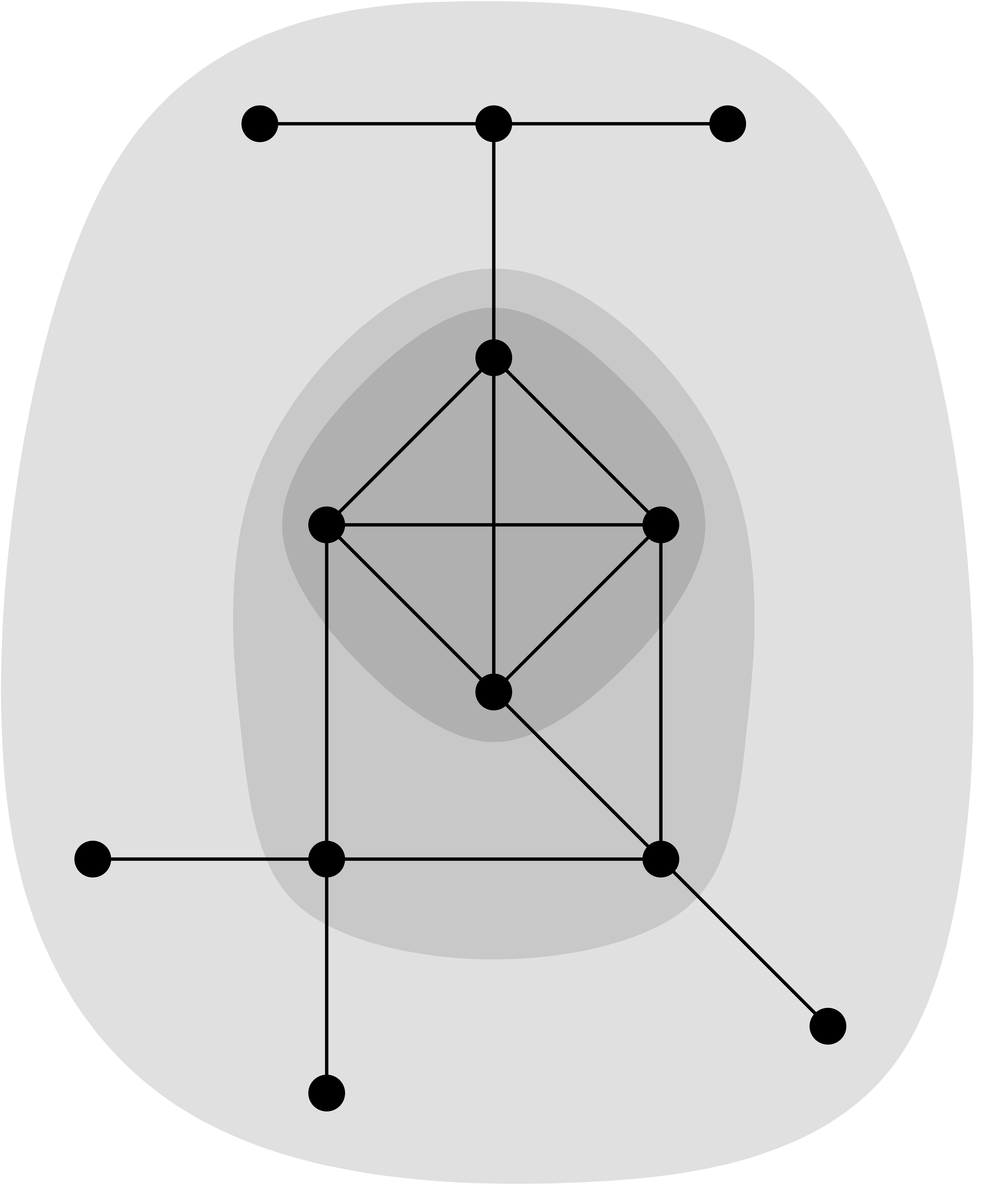
Groups of Nodes • Cores

For many purposes, a clique is too stringent a notion of grouping to be useful.

The *k-core* is a more flexible grouping notion.

By contrast with a clique, where each node is joined to all the others, **a *k*-core is a connected set of nodes where each is joined to at least *k* of the others**. Thus, in a 2-core, for instance, every node is joined to at least two others in the set.

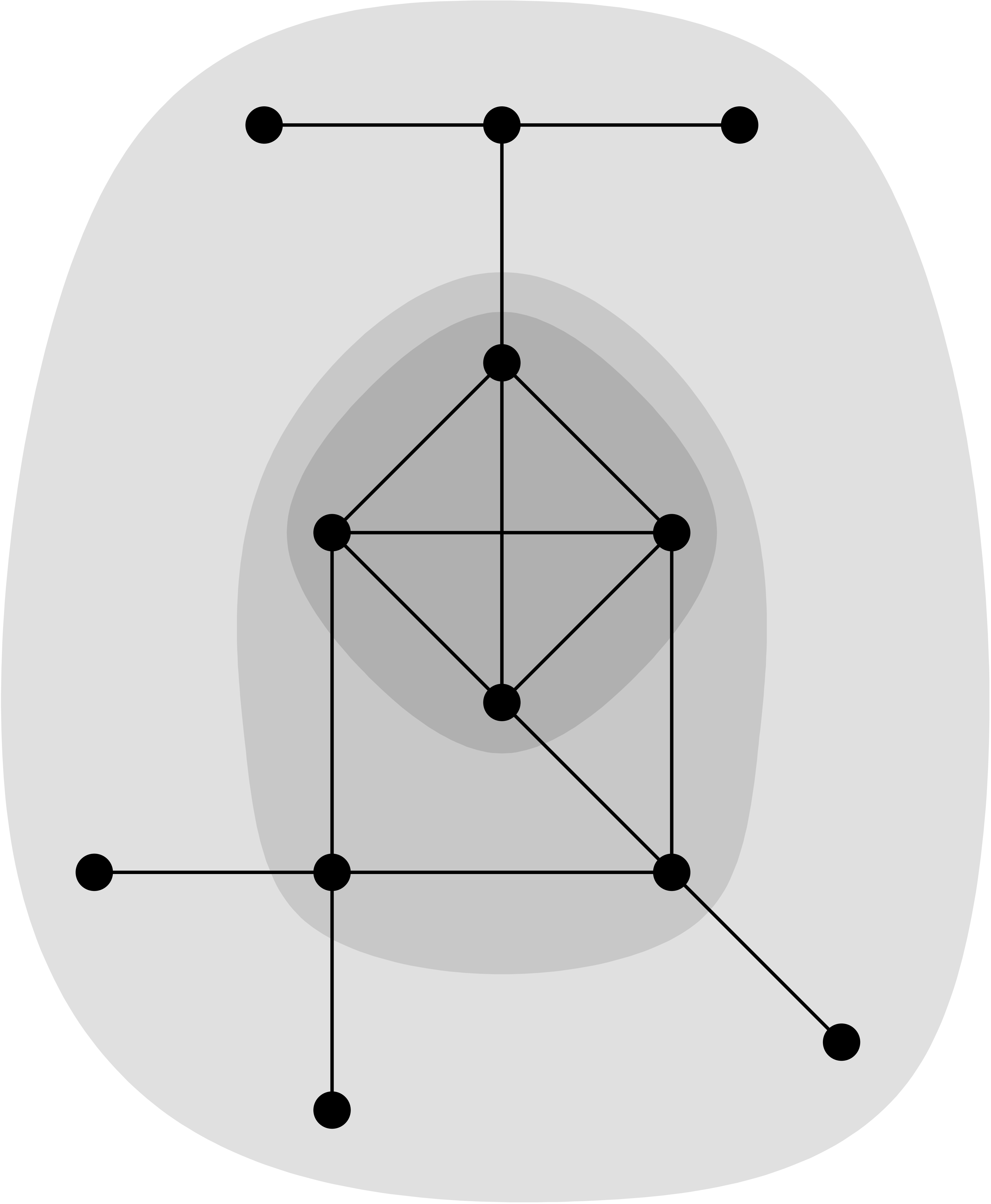
The *k*-core is not the only possible relaxation of a clique, but it is a particularly useful one for the very practical reason that *k*-cores are easy to find.



Groups of Nodes • Cores

A simple way to find k-cores is to start with a given network and remove from it any nodes that have degree less than *k*, along with their attached edges, repeating the process as long as there is a drop in degree between one passage and the other.

What is left over is, by definition, a *k*-core or a set of *k*-cores, since each node is connected to at least *k* others. Note that we are not necessarily left with a *single k*-core—there is no guarantee that the network will be connected once we are done pruning it, even if it was connected to start with.

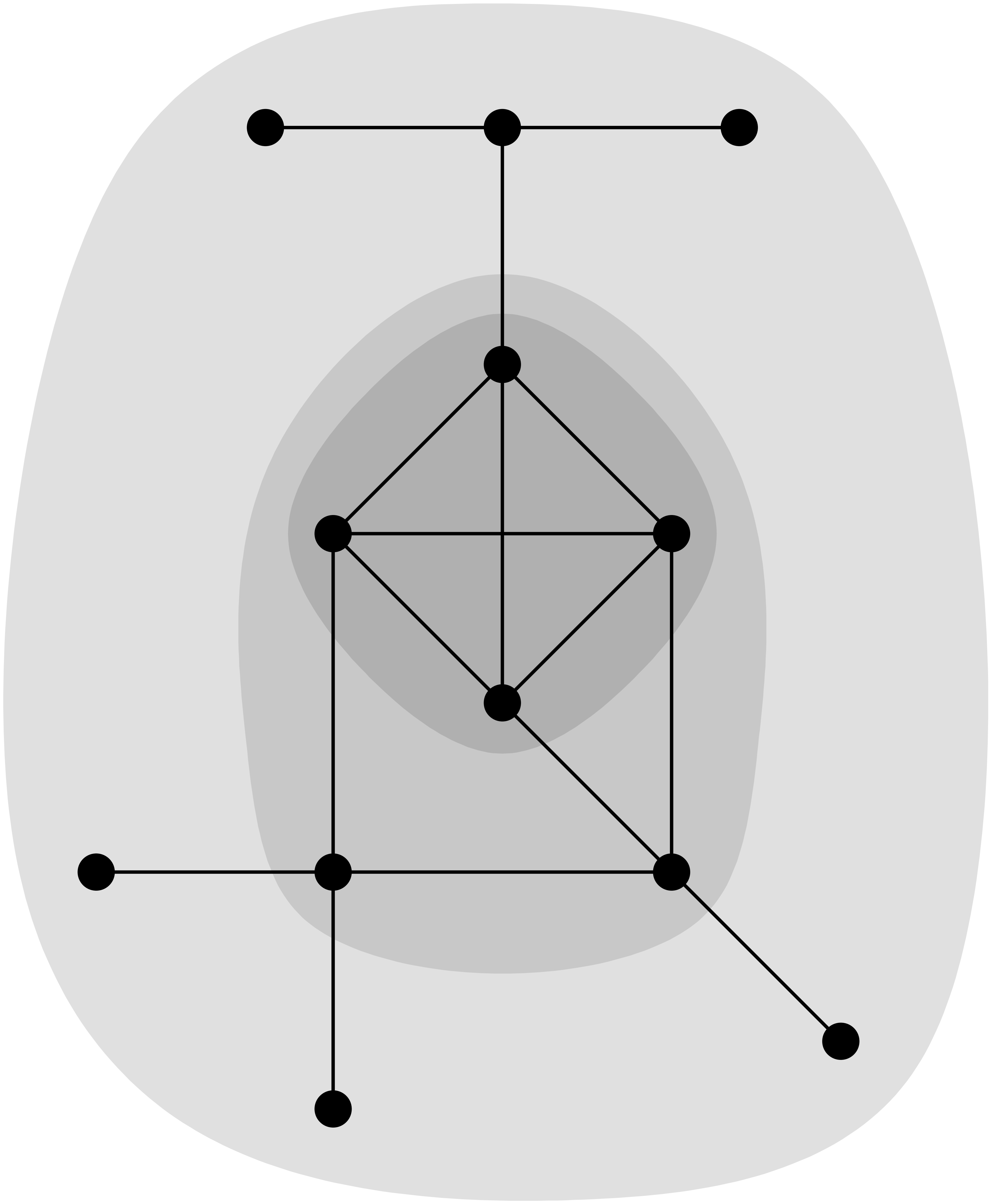


Groups of Nodes • Cores

The breakdown of a network into cores for all values of *k* provides an onion-like decomposition into layers within layers—1-, 2-, 3-cores, and so forth, culminating at the highest value of *k* for which cores exist.

This decomposition is sometimes used as a measure of *core–periphery structure* in networks: nodes that lie within the highest-*k* cores are "core" nodes within the network, while nodes outside those cores are "peripheral" nodes.

In this sense, the cores define a kind of centrality measure, and they are sometimes used that way.



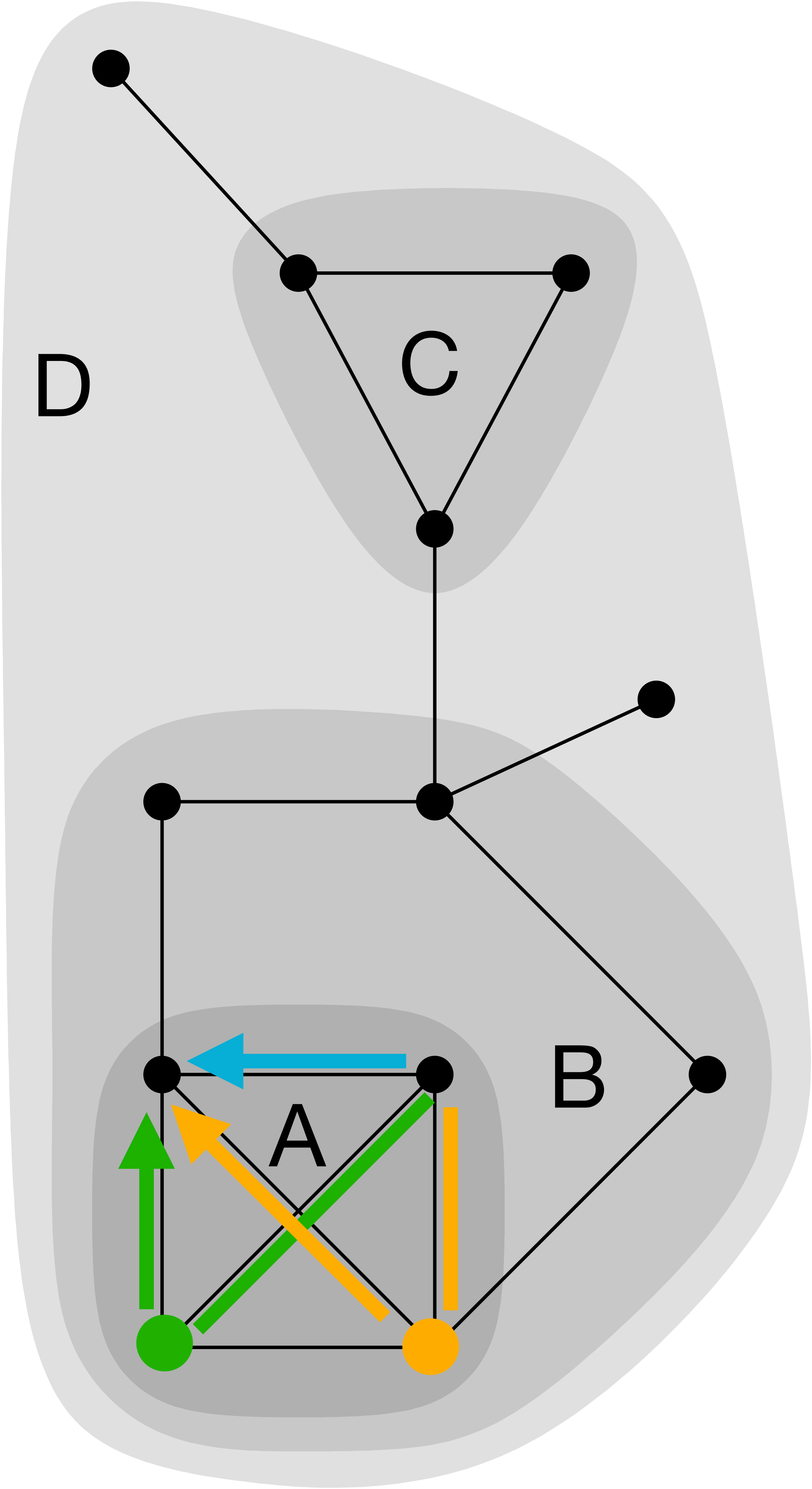
Groups of Nodes • Components and K-components

Reminder: a *component* in an undirected network is a (maximal) set of nodes, each with a path to each of the others.

A useful generalisation of this concept is the k-component. A *k-component* (sometimes also called a k-connected component) **is a set of nodes such that each is reachable from each of the others by at least *k node-independent paths*** (paths that do not share any node but the source and the target ones).

A 1-component is an ordinary component—there is at least one path between every pair of nodes—and, like *k*-cores, *k*-components are nested within each other.

In the example on the right, we find one 3-component (A), two 2-components (B, C) and one 1-component (D).

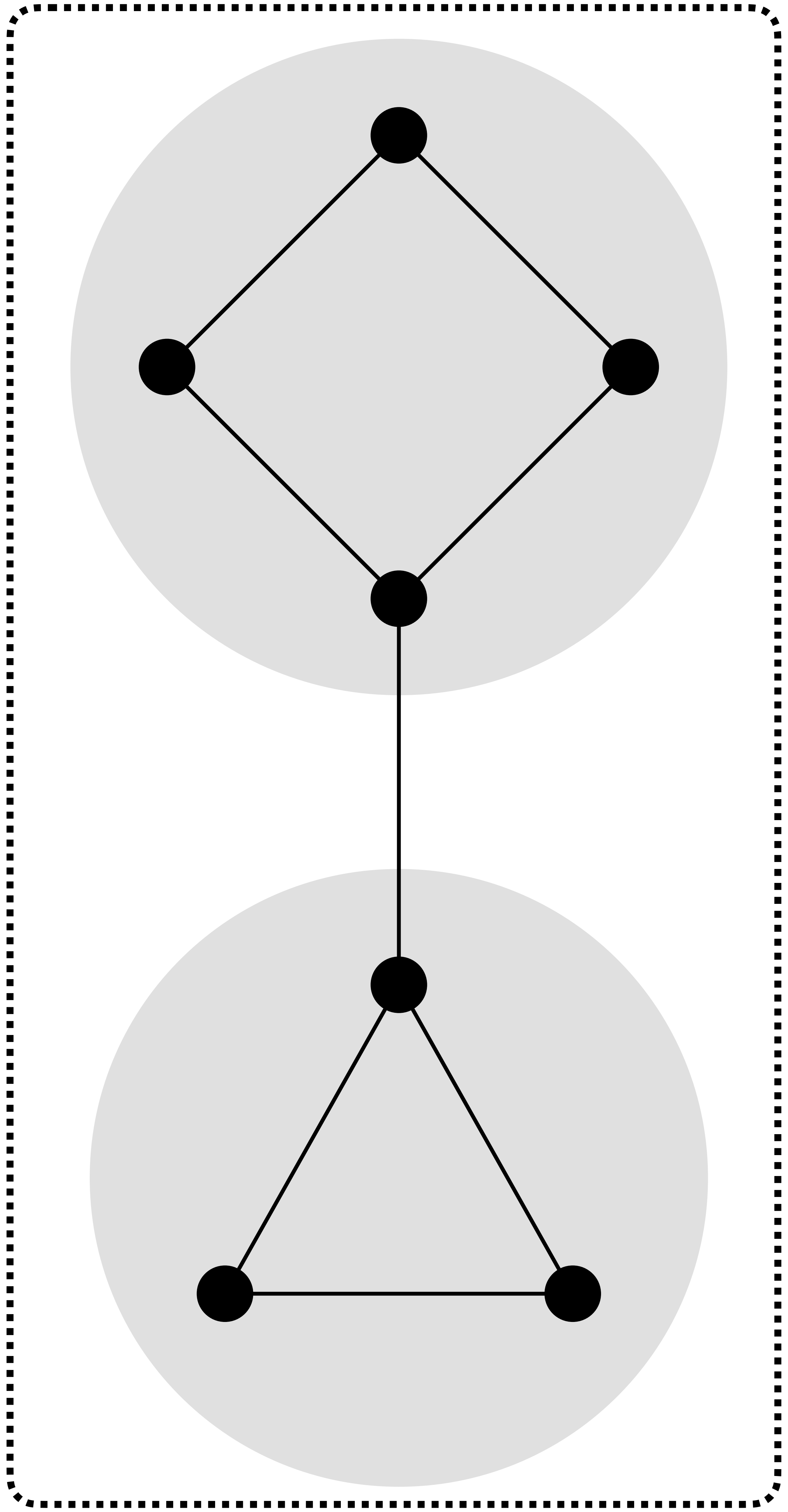


Groups of Nodes • Components and K-components

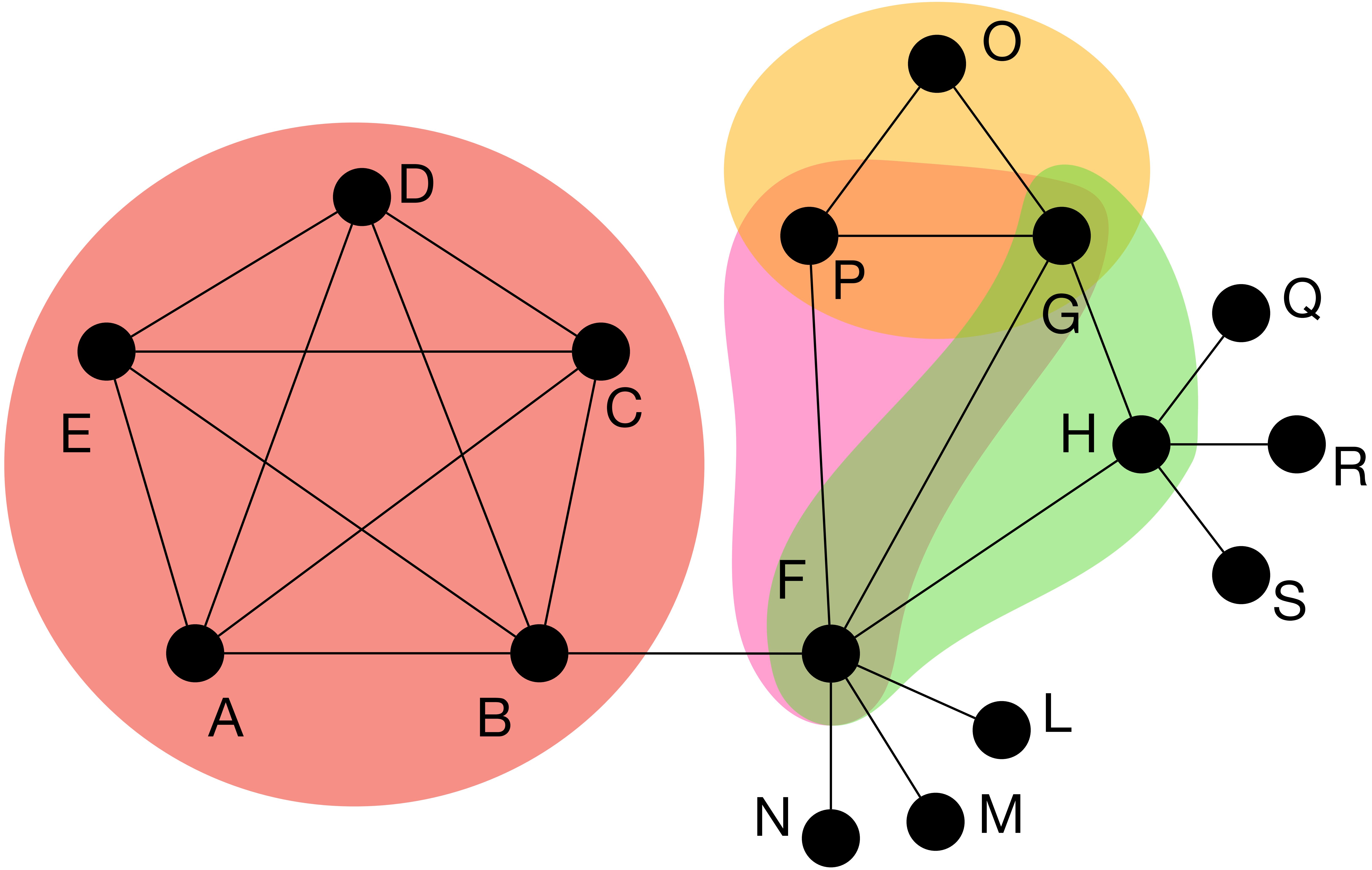
K-components might seem similar to *k*-cores, but there are important differences.

In the example on the right we find a single 2-core (represented by the dotted line) yet there are two separate 2-components in the network because the top-half and bottom-half of the network are connected by only one independent path in the middle, which separates the two 2-components.

In general, the number of node-independent paths between two nodes equals the number of nodes that we would need to remove to disconnect them. Indeed, this is an alternative way to define a *k*-component: a subset of a network in which no pair of nodes can be disconnected from each other by removing less than *k* other nodes.

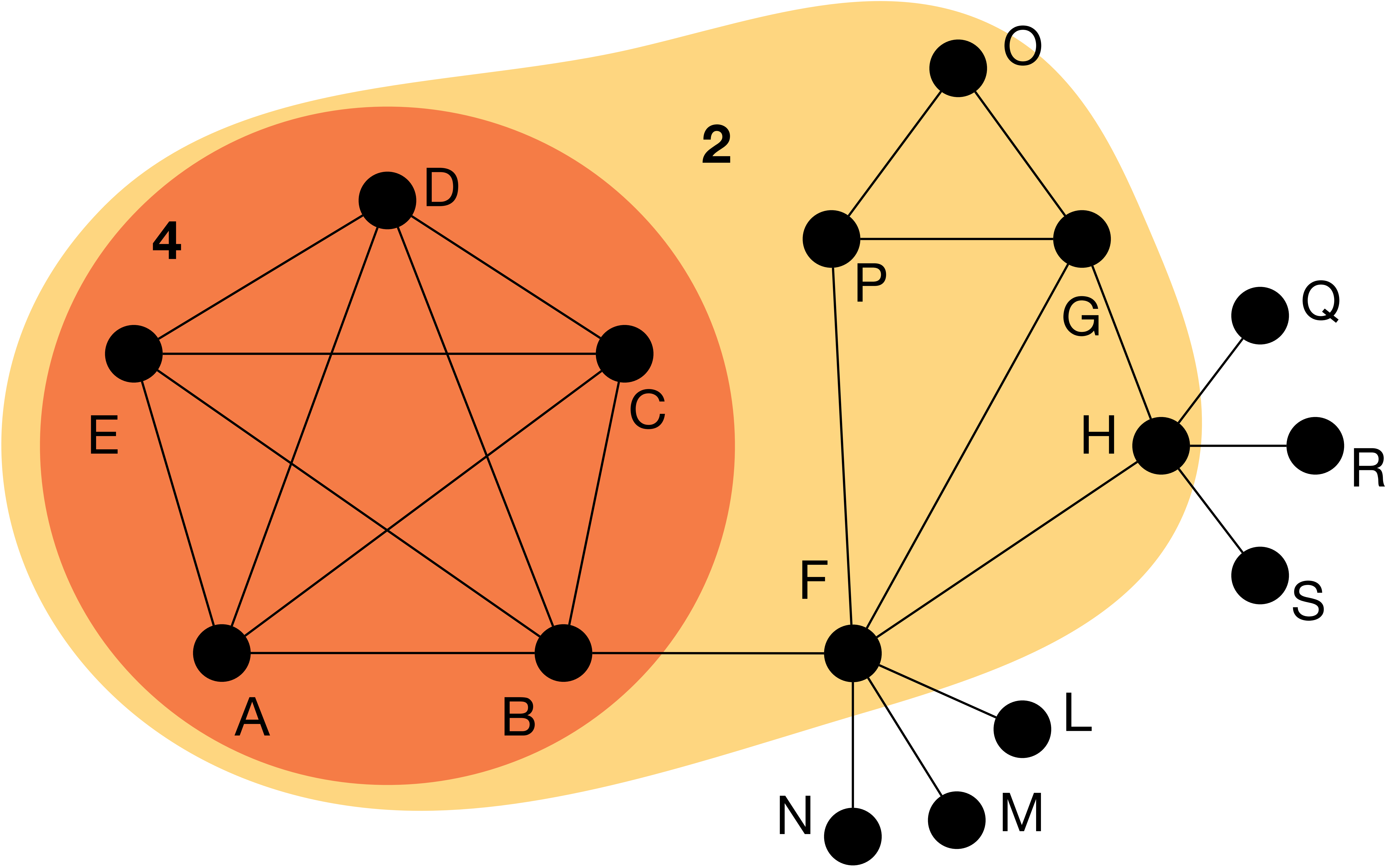


Groups of Nodes • Recap: Cliques, Cores, and Components



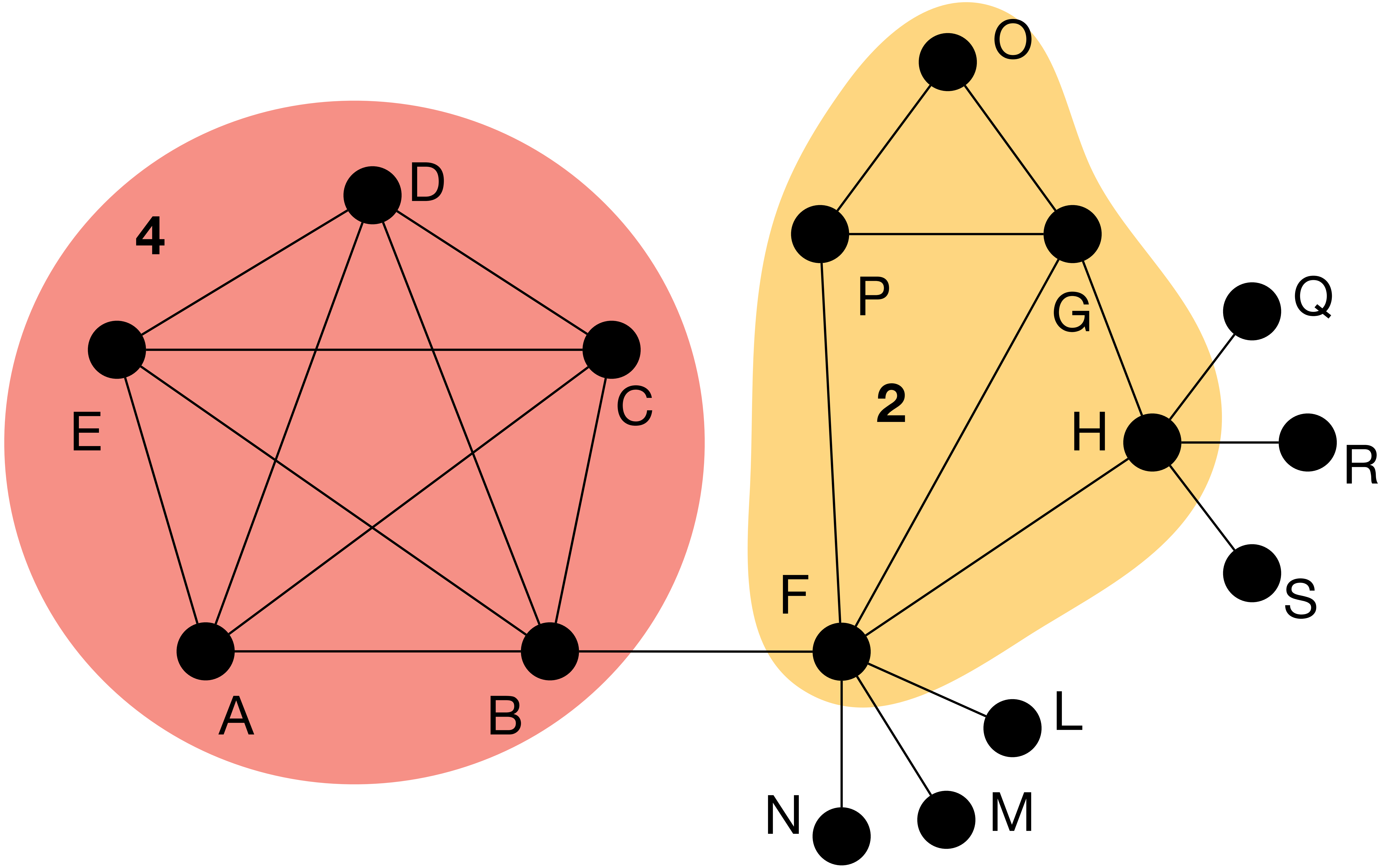
Clique: every member of the set is connected by an edge to every other member.

Groups of Nodes • Recap: Cliques, Cores, and Components



***k*-core:** a connected set of nodes where each is joined to at least *k* of the others

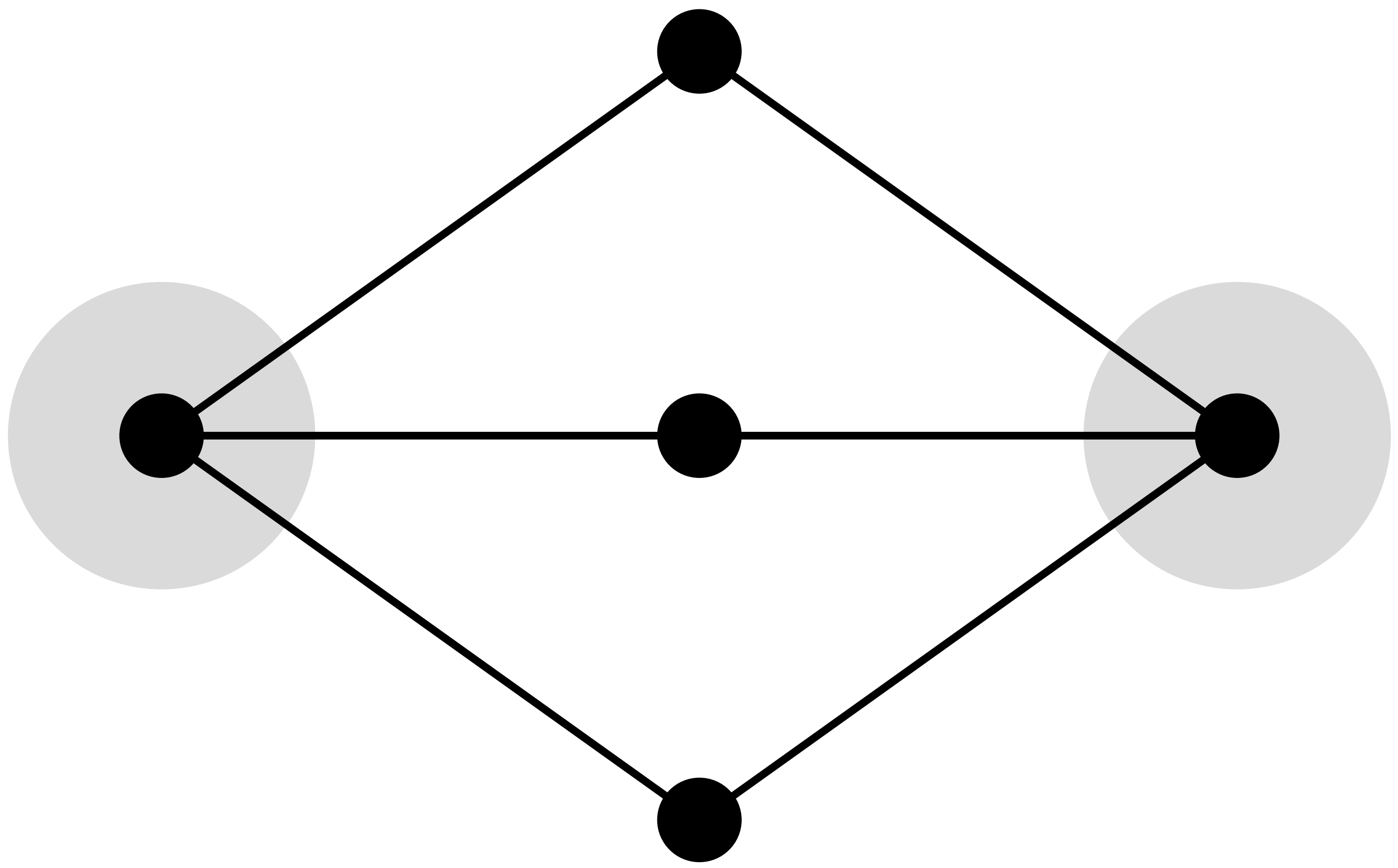
Groups of Nodes • Recap: Cliques, Cores, and Components



k-component: a set of nodes where each is reachable from each member by at least k unique paths.

Groups of Nodes • Alternatives to K-components

One disadvantage of *k*-components is that for *k* ≥ 3 they can be non-contiguous (e.g., the graph on the right).



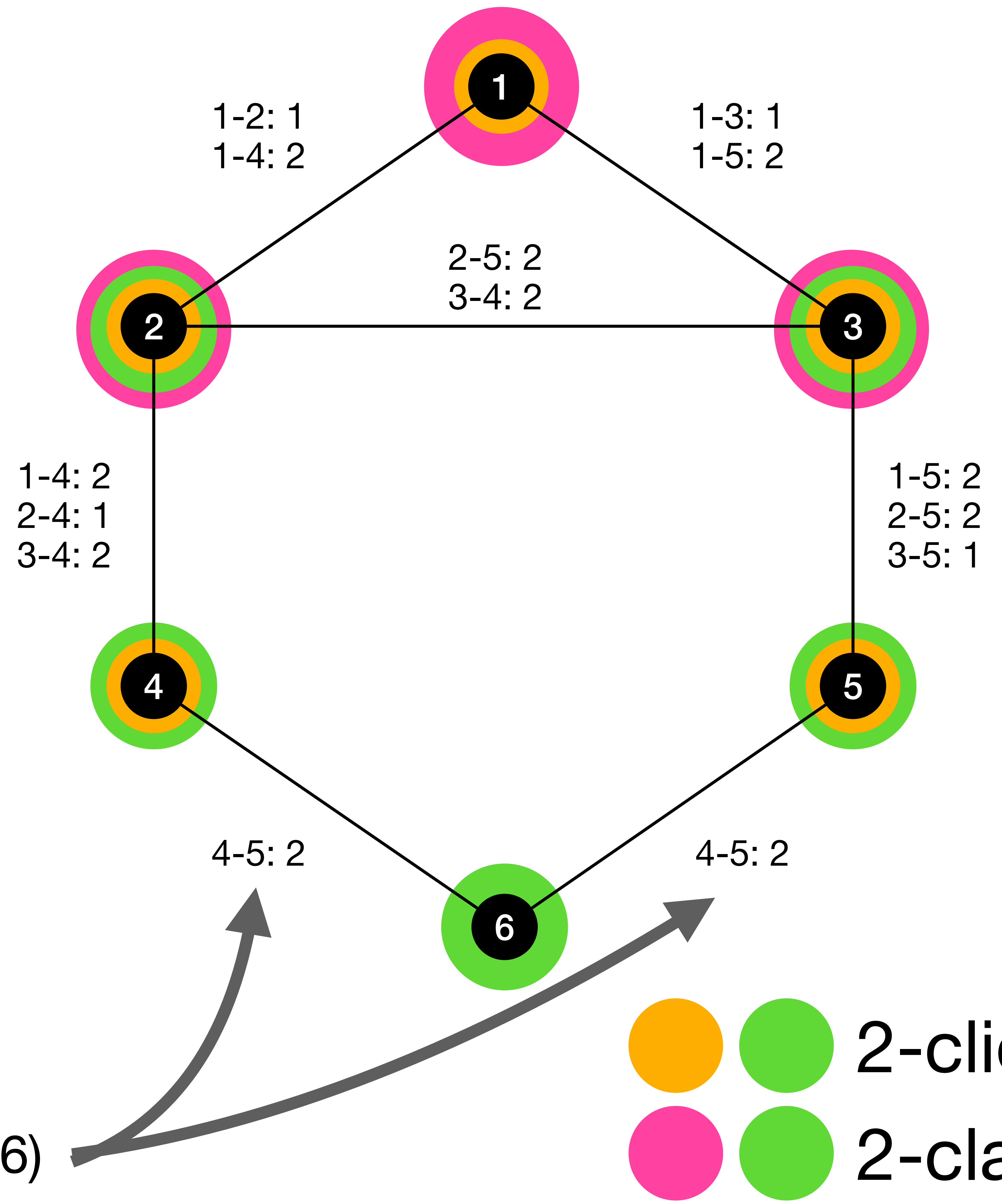
Sometimes, non-contiguous components are inappropriate to identify groups of nodes (imagine modelling different football teams whose grouping/structure is similar but for which it does not make sense to “combine” in the same group).

For this reason, researchers introduced alternative grouping definitions, like N-cliques, N-clans, K-plexes, and K-groups.

Groups of Nodes • Alternatives to K-components

N-cliques: generalisation of cliques that replace the strong constraint of the complete and maximum subgraph with the existence of a relationship between all the actors through a path of maximum length N.

N-clans: a restriction of N-cliques through the constraint that the longest path *in the group* is less than or equal to N. It corrects a defect in N-cliques that can form spurious groups by including “neighbouring” members that are (literally) closer to other groups.



🟠 uses a “spurious” path through a non-clique member (6)

🟠 🟢 2-cliques

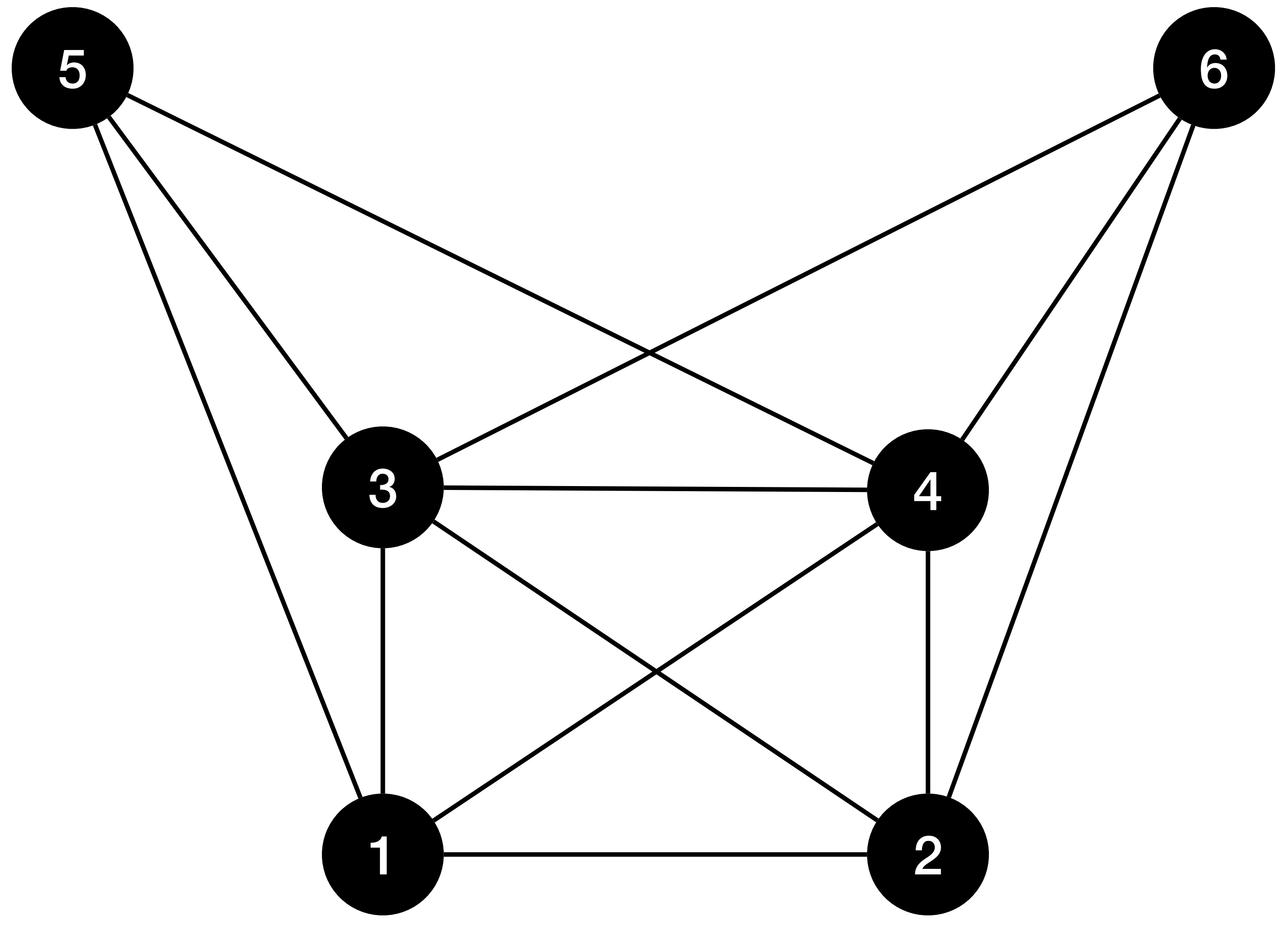
🔴 🟢 2-clans

Groups of Nodes • Alternatives to K-components

K-plexes: another generalisation of cliques that accepts as a member of the group any node that has at least $n - k$ links with the other nodes, where n is the total number of nodes that make up the group.

For example, A would be part of a 2-plex consisting of nodes B, C and D if it had a link with both B and C, but not with D, being D in turn linked to both B and C.

K-plexes generate many more smaller groups than the previous methods.



1-plexes (deg. case of clique): { 1, 2 , 3 ,4 }

2-plexes: 1-plexes ∪ { 1, 2, 3, 4, 6 }, { 1, 2, 3, 4, 5 }

3-plexes: 2-plexes ∪ { 1, 2, 3, 4, 5, 6 }

Groups of Nodes • Transitivity and Clustering Coefficients

In mathematics a relation $\mathscr{R}$ is said to be transitive if $a\ \mathscr{R}\ b$ and $b\ \mathscr{R}\ c$ together imply $a\ \mathscr{R}\ c$. In networks, if $\mathscr{R}$ is "connected by an edge" and $\mathscr{R}$ is transitive, we would have that "if a and b are connected and b and c are connected, then a and c are connected".

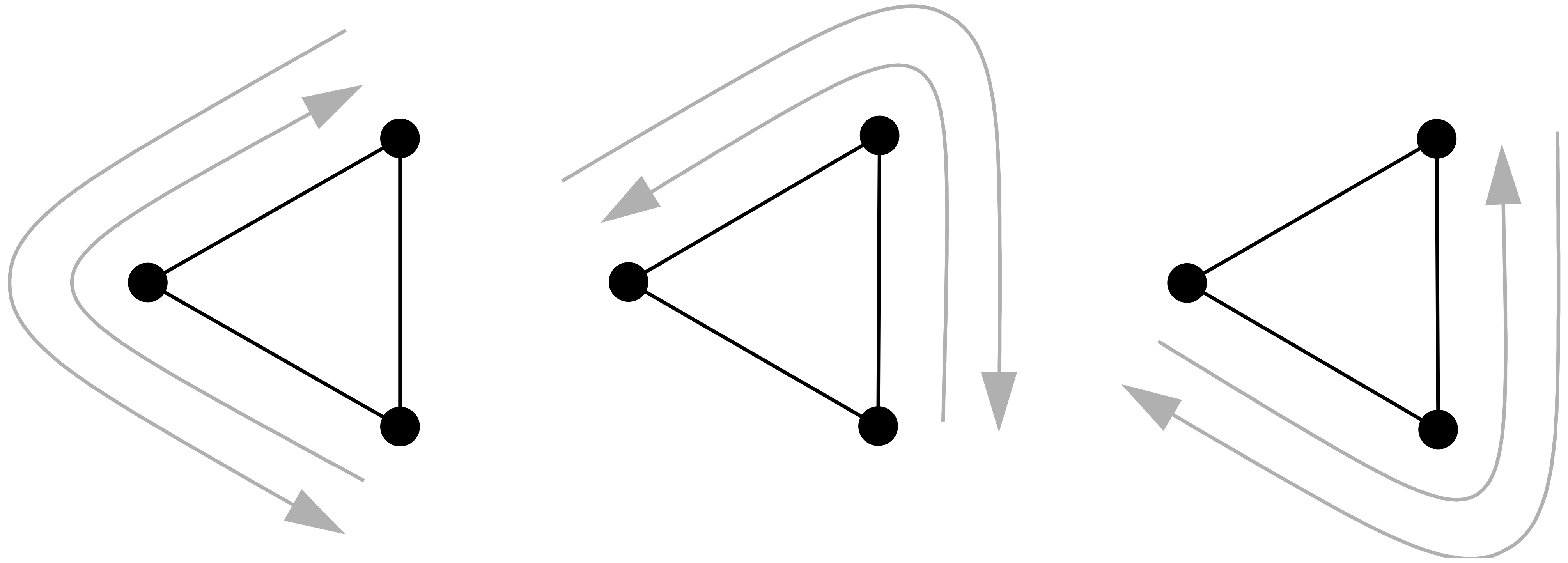
Perfect transitivity holds in a network when the network is a clique (and its graph is *complete*). *Partial transitivity* instead can indicate the *tendency* to extend that (missing) relation, e.g., if a and b are friends and b and c are friends, that does not guarantee that a and c are friends, however it makes it *likely*.

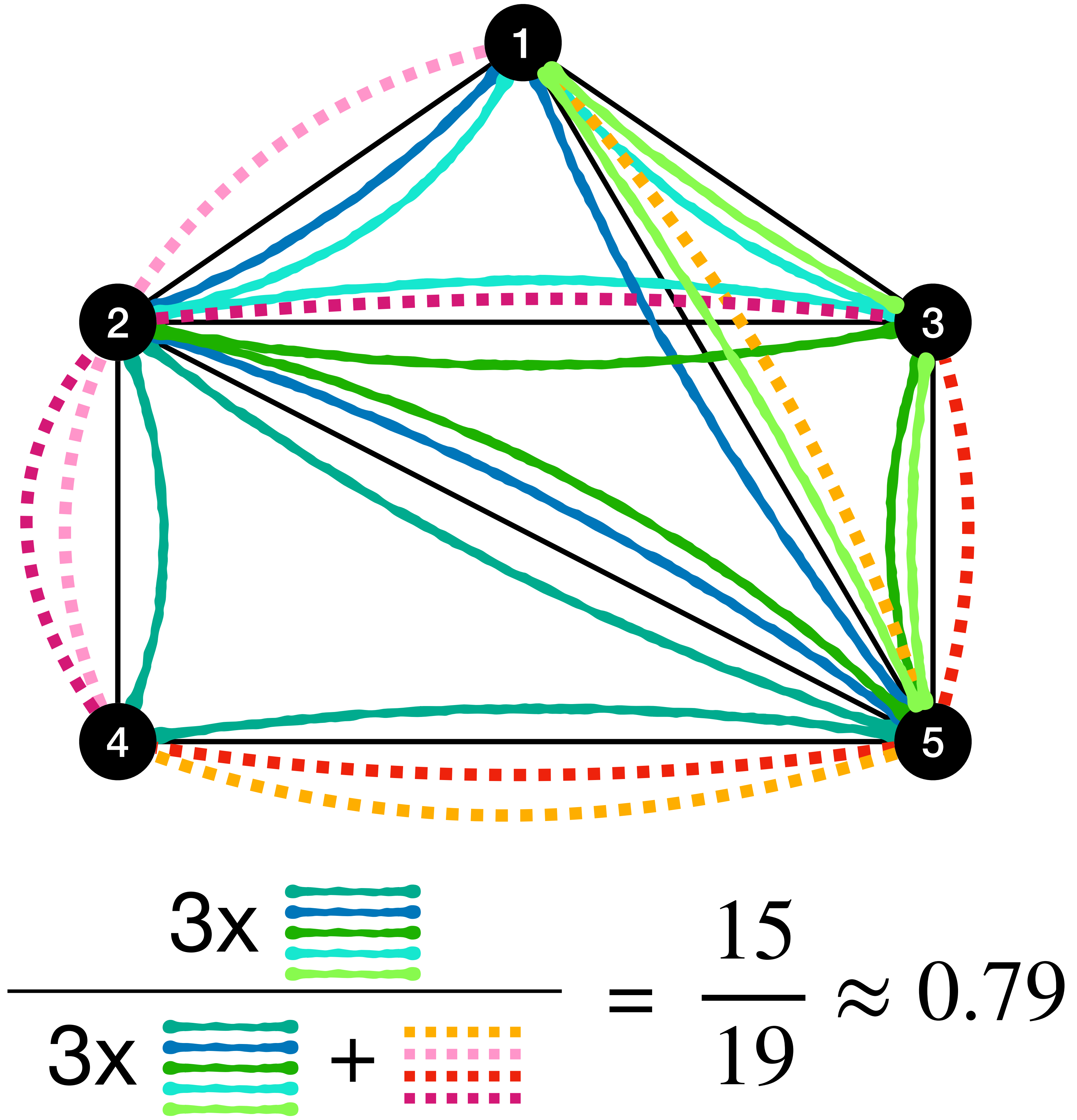
Transitivity is a property of triads that characterise different network structural configurations: *isolation* (when the triad is disconnected), *dyad* (when only two out of three nodes are in $\mathscr{R}$), *structural hole* (when the three nodes are in $\mathscr{R}$ except one dyad), *cluster* (when the triad enjoys perfect transitivity). Clusters are also called *closed triads* as they form 2-edge long paths among the members of the triad, closed by a third edge.

Groups of Nodes • Transitivity and Clustering Coefficients

The *clustering coefficient* is the fraction of paths of length two in the network that are closed. That is, we count all paths of length two, we count how many of them are closed, and then we divide the second number by the first to get a clustering coefficient *C* that lies in the range from zero to one:

$$C = \frac{\text{number of closed paths of length two}}{\text{number of paths of length two}}$$



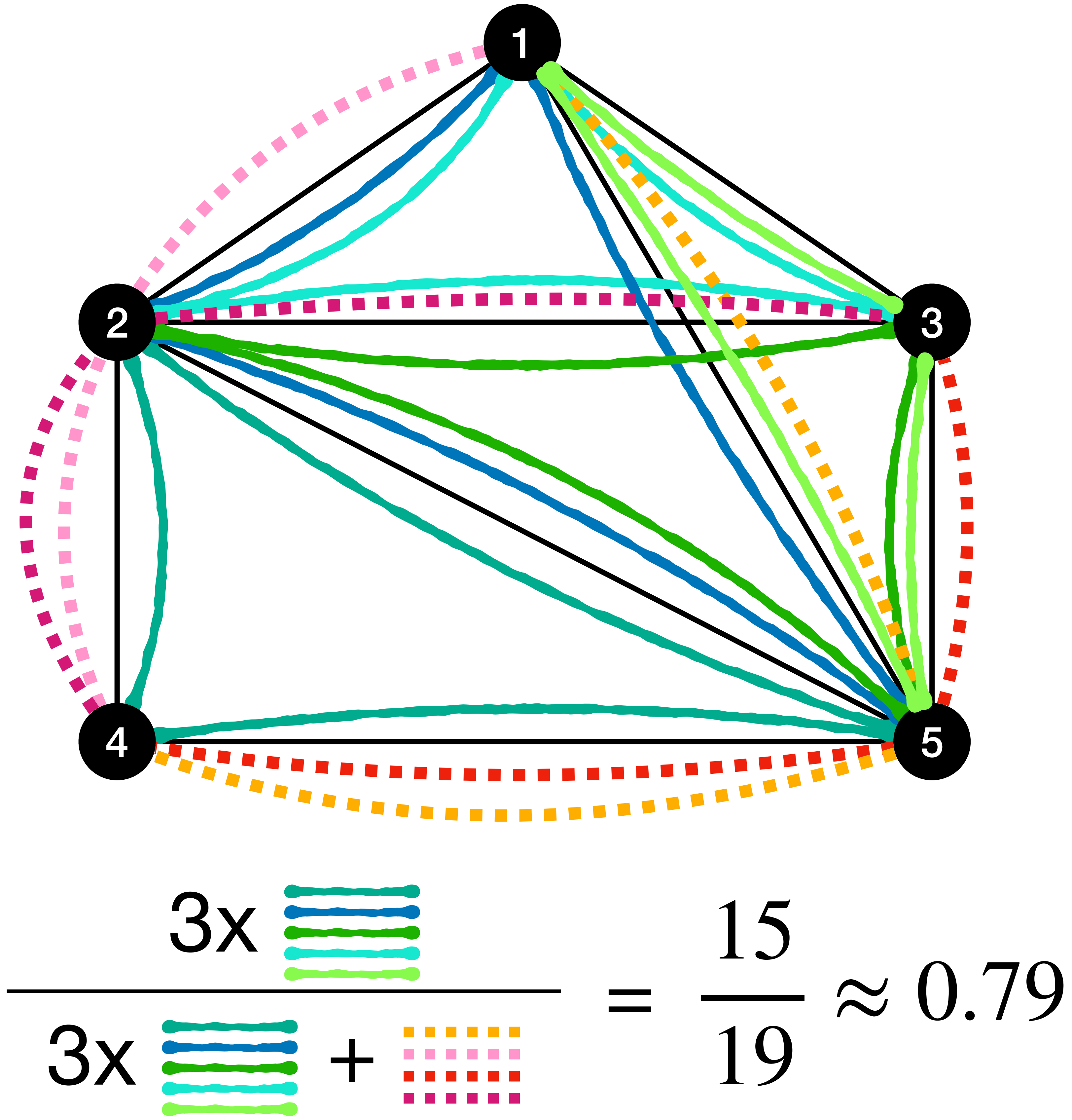


Groups of Nodes • Transitivity and Clustering Coefficients

The *clustering coefficient* is the fraction of paths of length two in the network that are closed. That is, we count all paths of length two, we count how many of them are closed, and then we divide the second number by the first to get a clustering coefficient *C* that lies in the range from zero to one:

$$C = \frac{\text{number of closed paths of length two}}{\text{number of paths of length two}}$$

C = 1 implies perfect transitivity. *C* = 0 implies no closed triads. For reference, e.g., a network of who-sends-email-to-whom in a large university had *C* = 0.16. Technological and biological networks tend to have lower values, e.g., the Internet has a clustering coefficient of ~0.01.



Groups of Nodes • Local Clustering and Redundancy

While the clustering coefficient is a property of an entire network, it also useful to define a local clustering coefficient C_i for a single node *i:*

$$C_i = \frac{\text{number of pairs of neightbours of } i \text{ that are connected}}{\text{number of pairs of neighbours of } i}$$

Hence, to calculate C_i we go through all distinct pairs of nodes that are neighbours of *i*, count the number of such pairs that are connected to each other, and divide by the total number of pairs (having d_i being the degree of the node, the total number of pairs corresponds the binomial coefficient $d_i(d_i - 1)/2$).

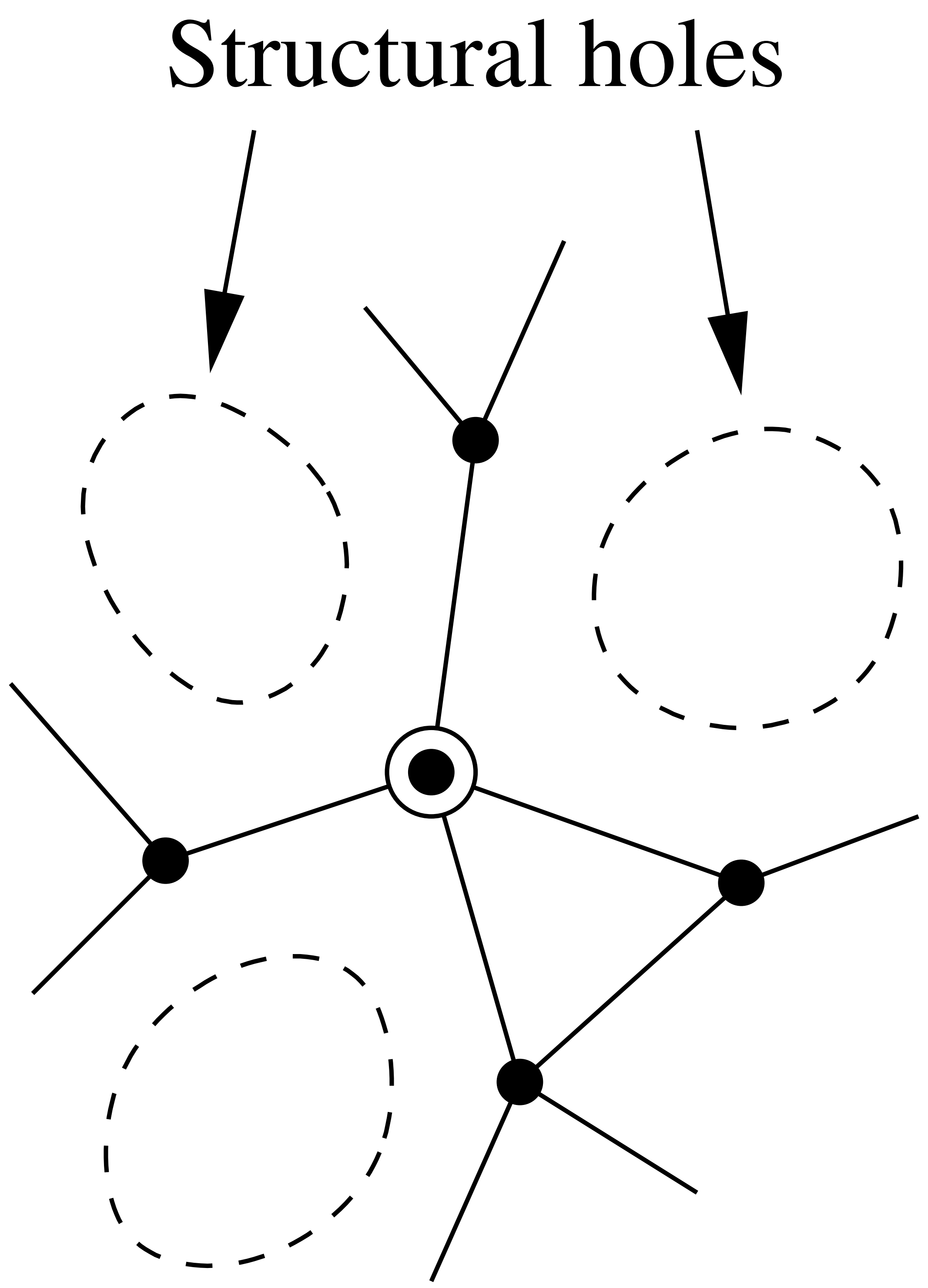
The *local clustering coefficient* represents the average probability that a pair of nodes related to it by $\mathscr{R}$ are also in $\mathscr{R}$ with each other. Since for nodes with degree zero or one the number of pairs of neighbours is zero and C_i would be not well defined, by convention $C_i = 0$ for those cases.

Groups of Nodes • Local Clustering and Redundancy

Local clustering can be used as an indicator of *structural holes* in a network.

Structural holes are interesting for a number of reasons, depending on the context of the research.

E.g, in a transport/information network structural holes are an issue because they represent missing alternative routes in the network. Contrarily, if we model the spread of a pandemic, structural holes work as barriers to the diffusion of the disease. Structural holes can also represent power for a node whose neighbours lack connections, as those missing links give control over information flow between those neighbours.

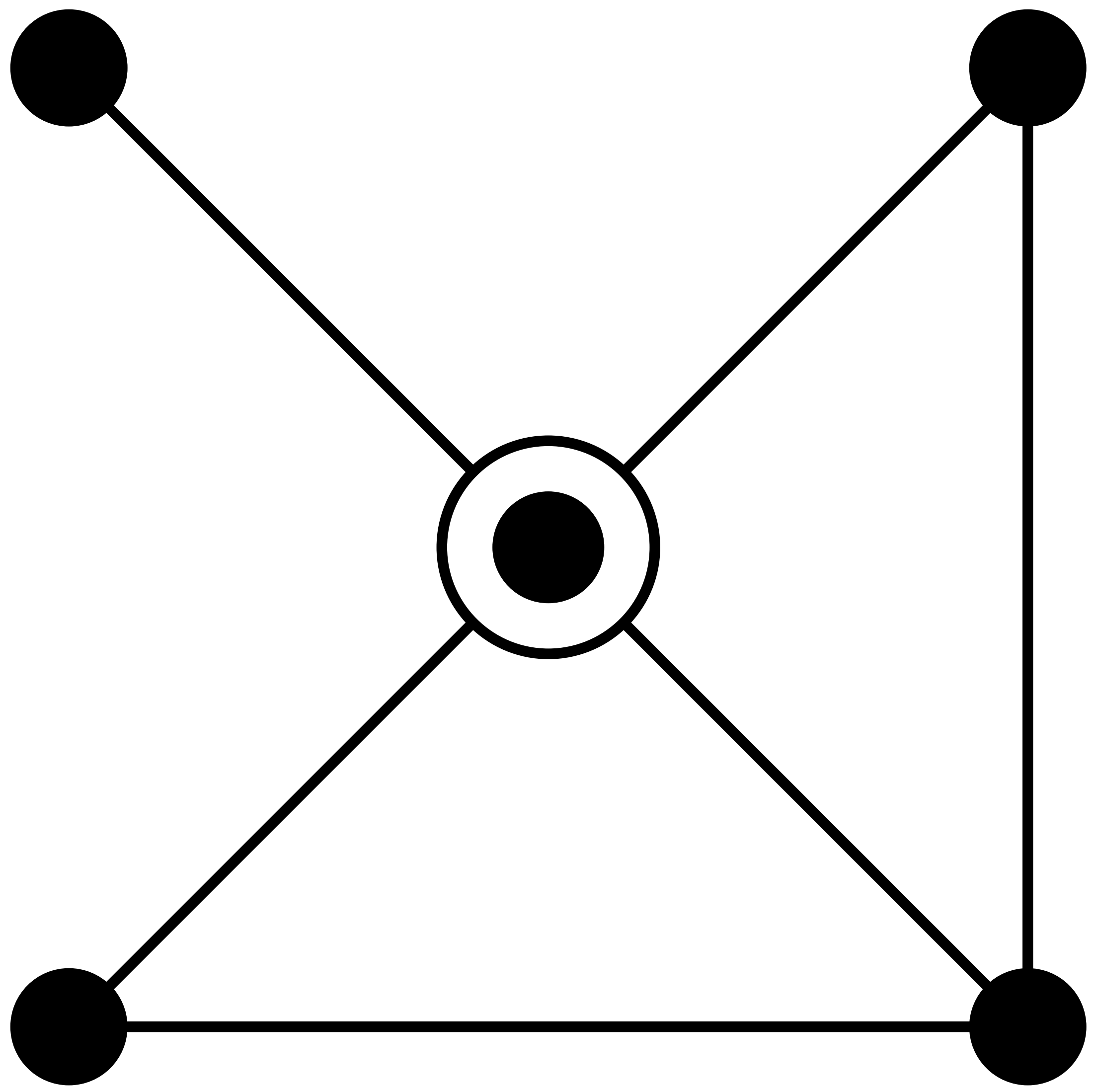


Thus, local clustering can be seen as a type of centrality measure, where the smaller the values the more "powerful" the node.

Groups of Nodes • Local Clustering and Redundancy

Local clustering is (also historically) strictly linked to the concept of *redundancy*, whose definition R_i of a node *i* corresponds to the **average number of connections from a neighbour of *i* to the other neighboursof *i*.**

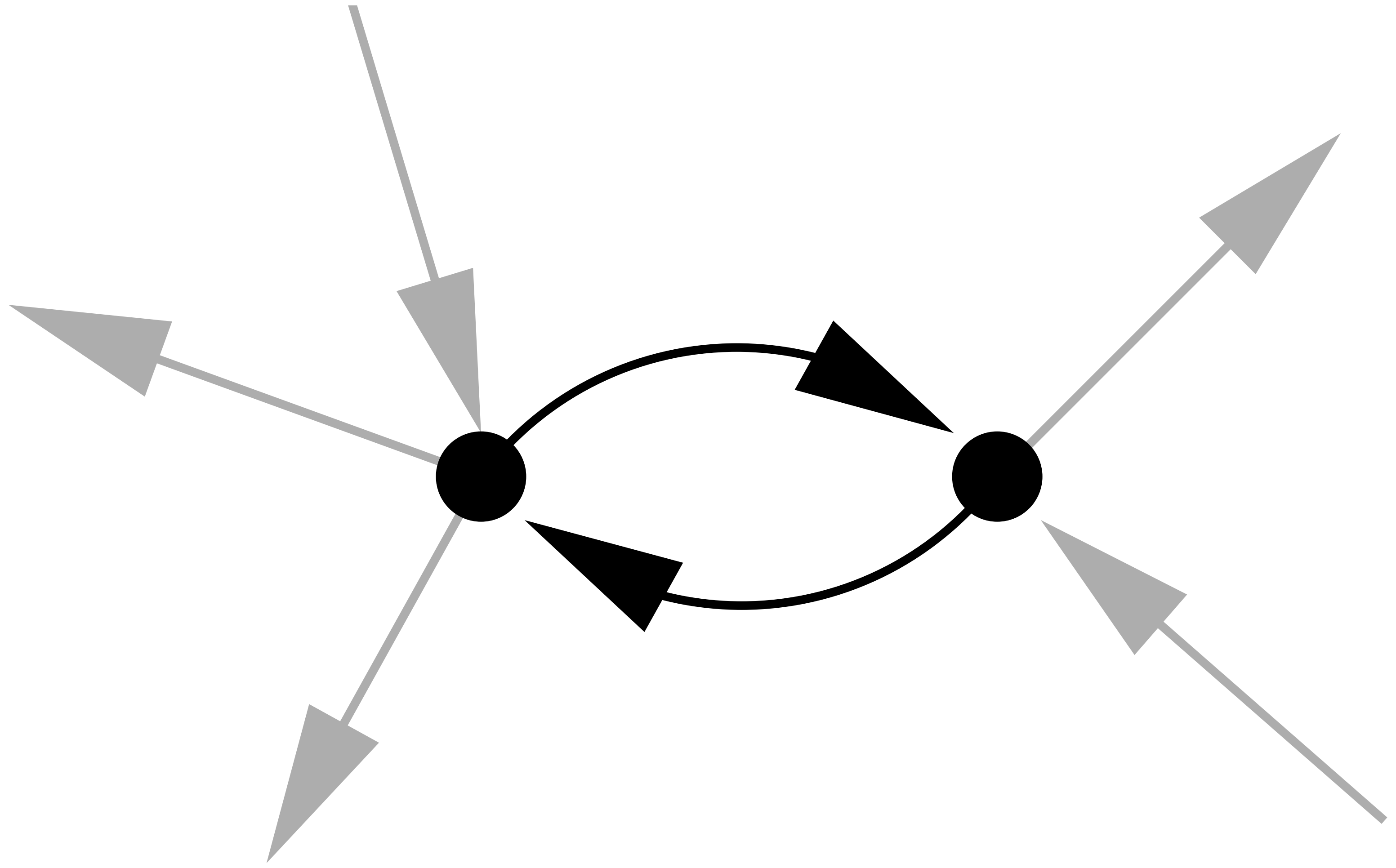
For example, in the graph on the right, the central node has four neighbours and each of those four *could* be acquainted with any of the three others, but in this case none of them is connected to all three. One is connected to none of the others, two are connected to one other, and the last is connected to two others. The redundancy of the central nodes is therefore (0+1+1+2)/ 4 = 1



The minimum possible value of the redundancy of a node *i* is zero and the maximum is $d_i - 1$, where d_i is the degree of the node.

Reciprocity

While the clustering coefficient focuses on triads (being the fundamental, shortest loop), depending on the context focussing on tetrads, pentads or more closed groups can be of interest.



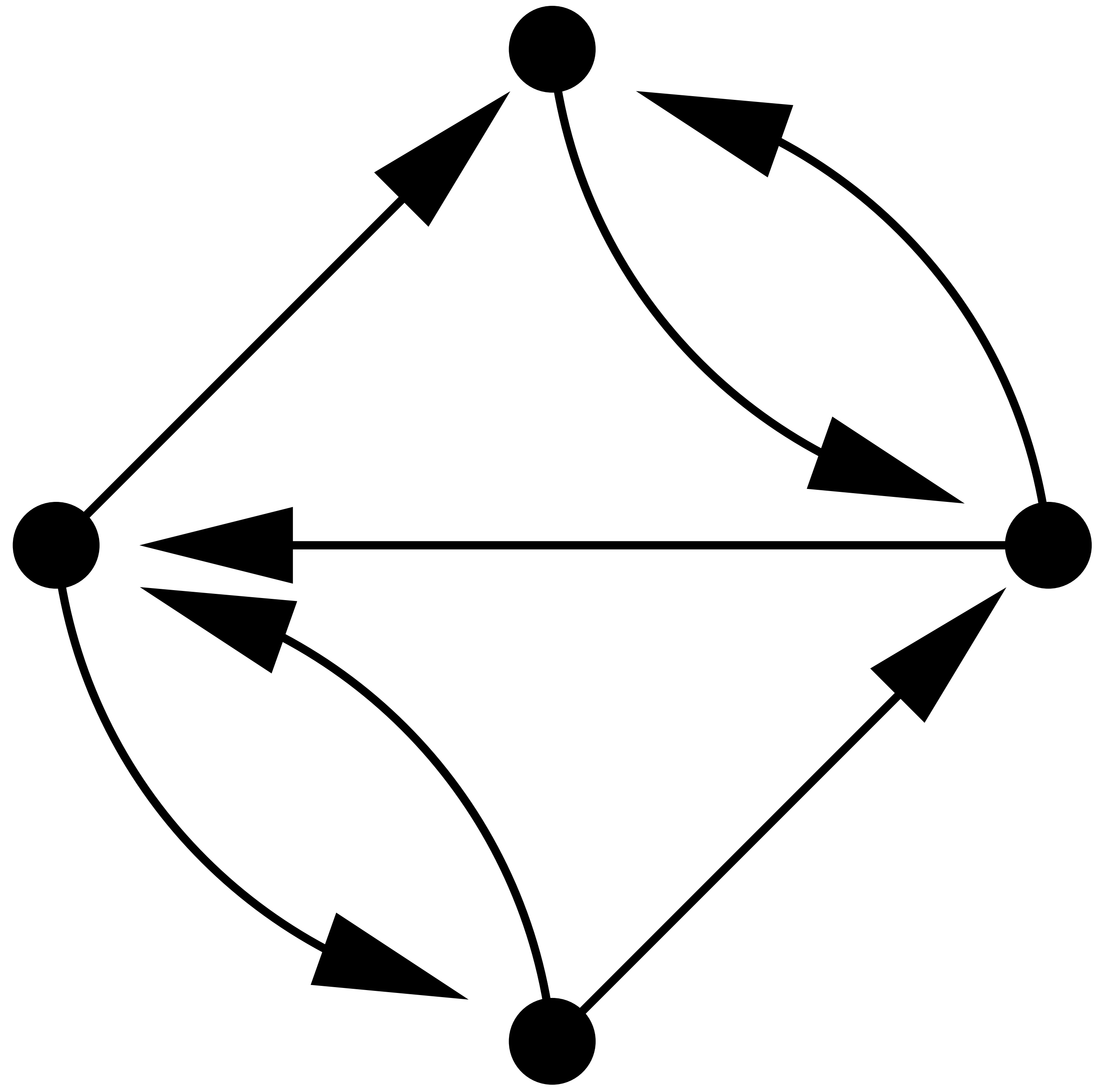
For example, in a *directed* network we can have loops of length two and it is interesting to ask about the frequency of occurrence of these loops also.

The frequency of loops of length two is measured by the *reciprocity*, which estimates how likely it is that two nodes point at each other. If there is a directed edge from node *i* to node *j* in a directed network and there is also an edge from *j* to *i,* then we say the edges are *reciprocated*. Let *m* be the total number of edges in the network, reciprocity is:

$$r = \frac{1}{m} \sum_{ij} A_{ij} \ A_{ji}$$

Reciprocity

For example, in the graph on the right there are seven directed edges and four of them are reciprocated, so the reciprocity is $r = 4/7 \simeq 0.57$. That value is about the same seen on the World Wide Web, where about 57% of web pages link back to a web page that points to it.



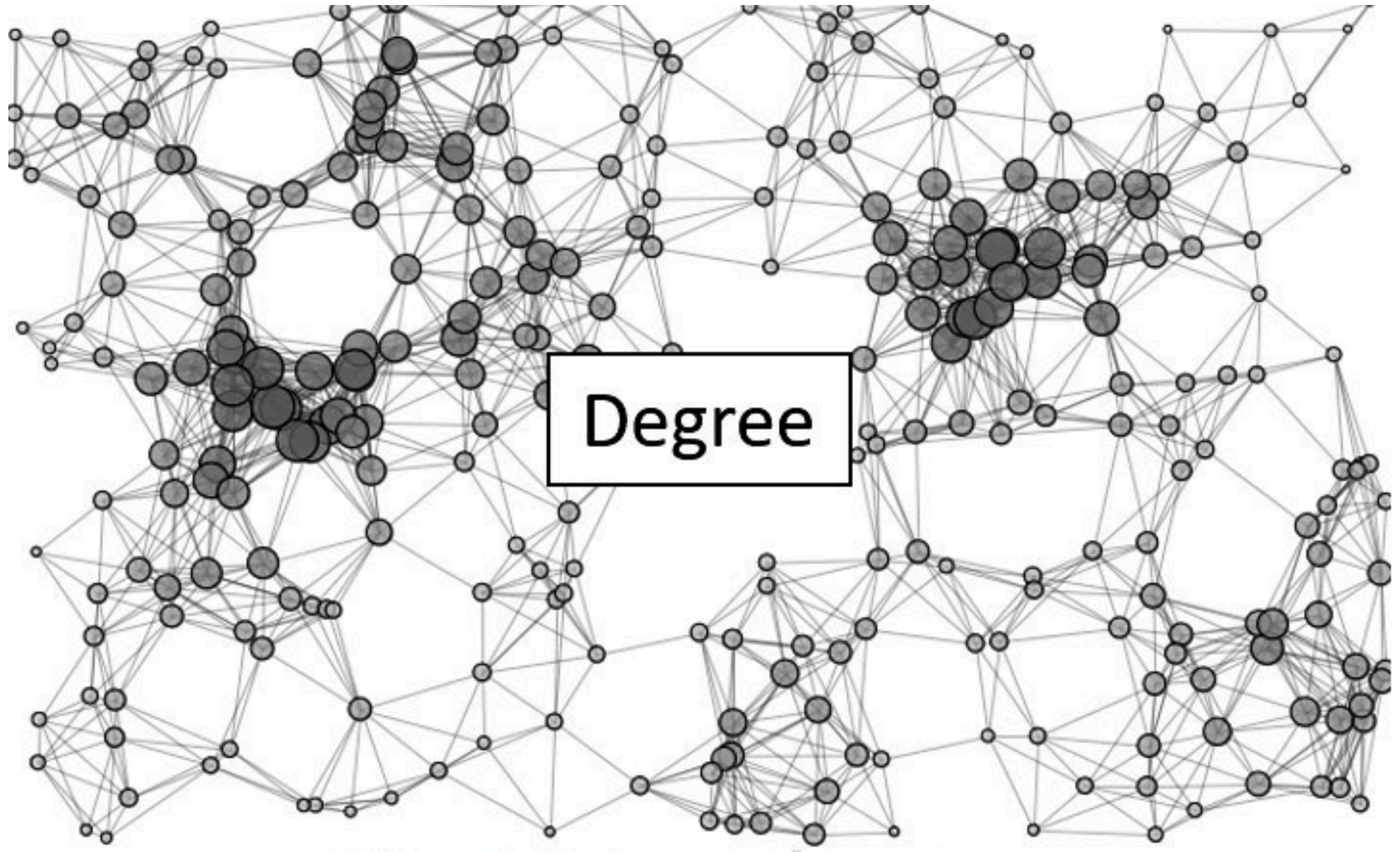
As another example, in a network of who-has-whom in their email address book, it was found that the reciprocity was about $r = 0.23$, while in a study of friendship networks from a large set of US high schools estimated a reciprocity between 0.3 to 0.5, depending on the school.

$$r = \frac{1}{m} \sum_{ij} A_{ij}\ A_{ji}$$

A Visual Wrap-up

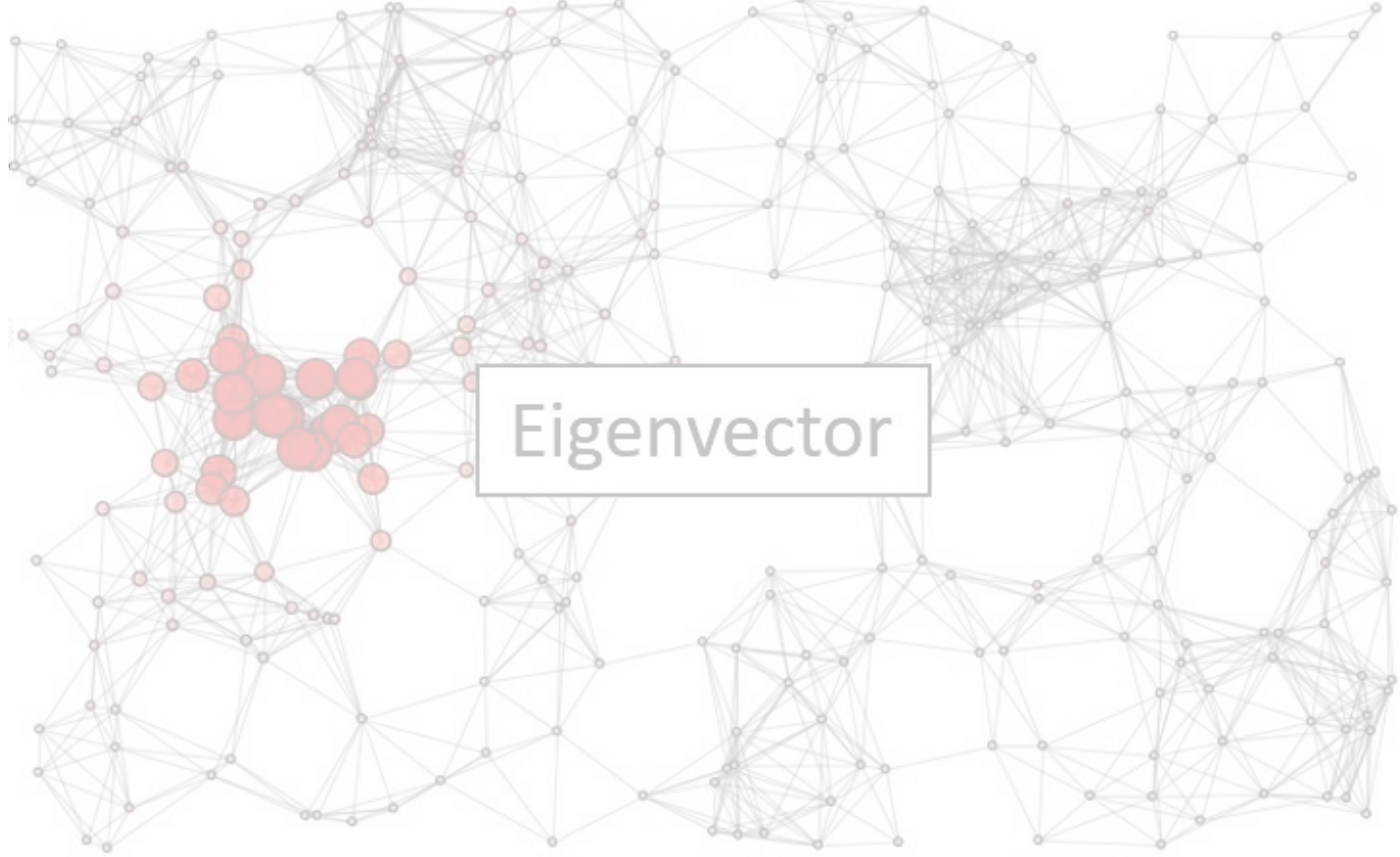
Degree
Betweenness
Clustering
Eigenvector
Closeness
Page Rank

A Visual Wrap-up

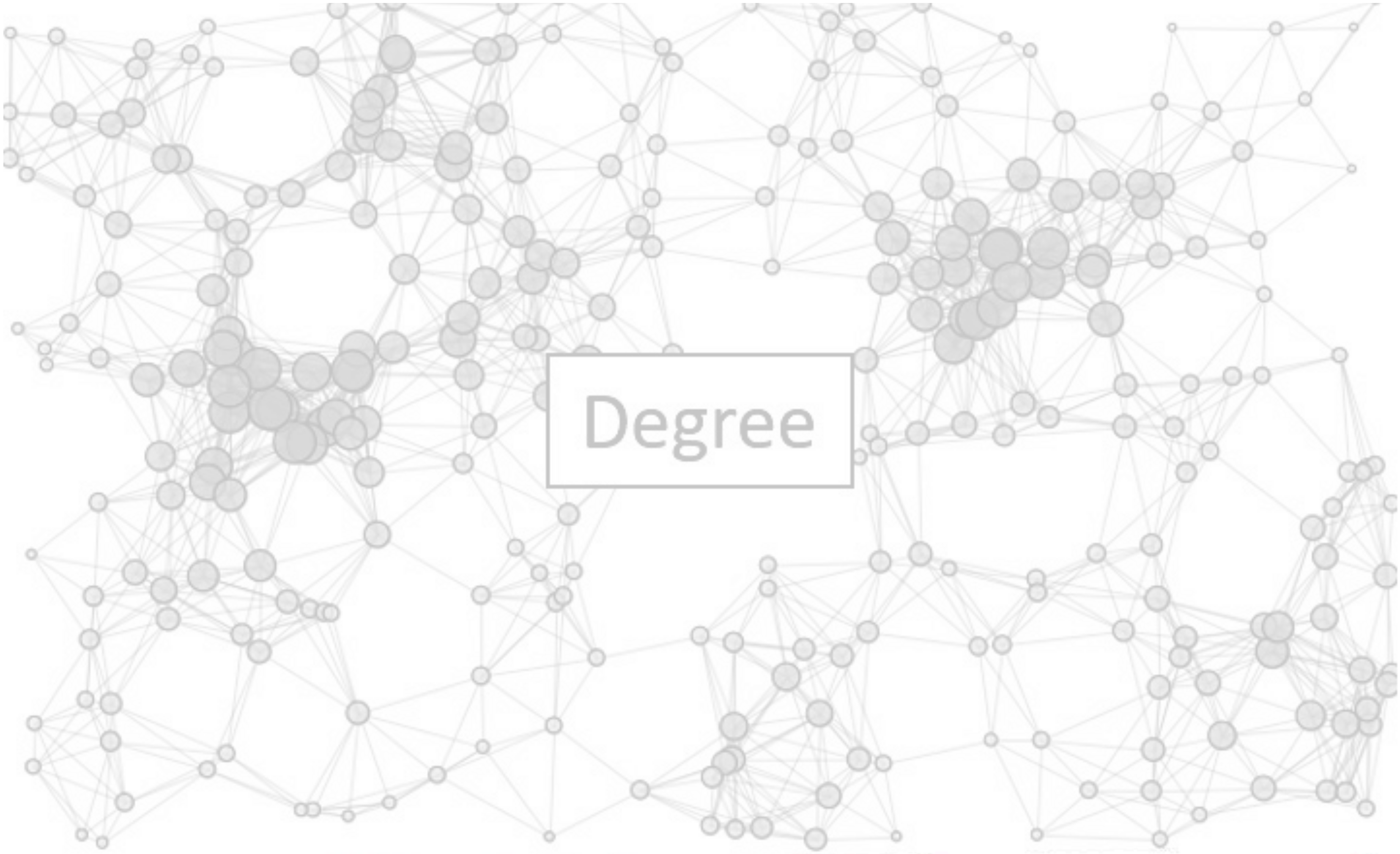


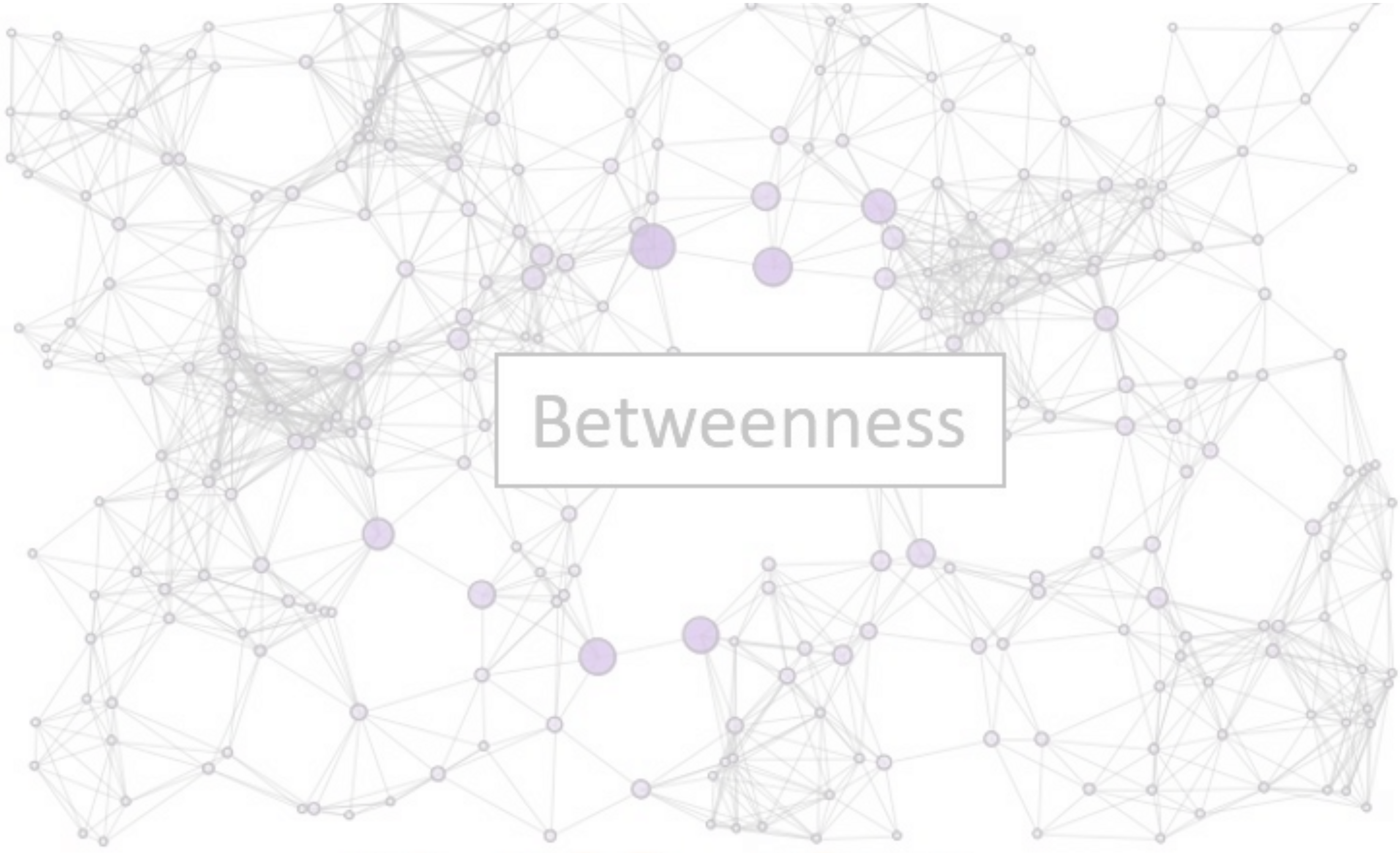
the number of connections to a node

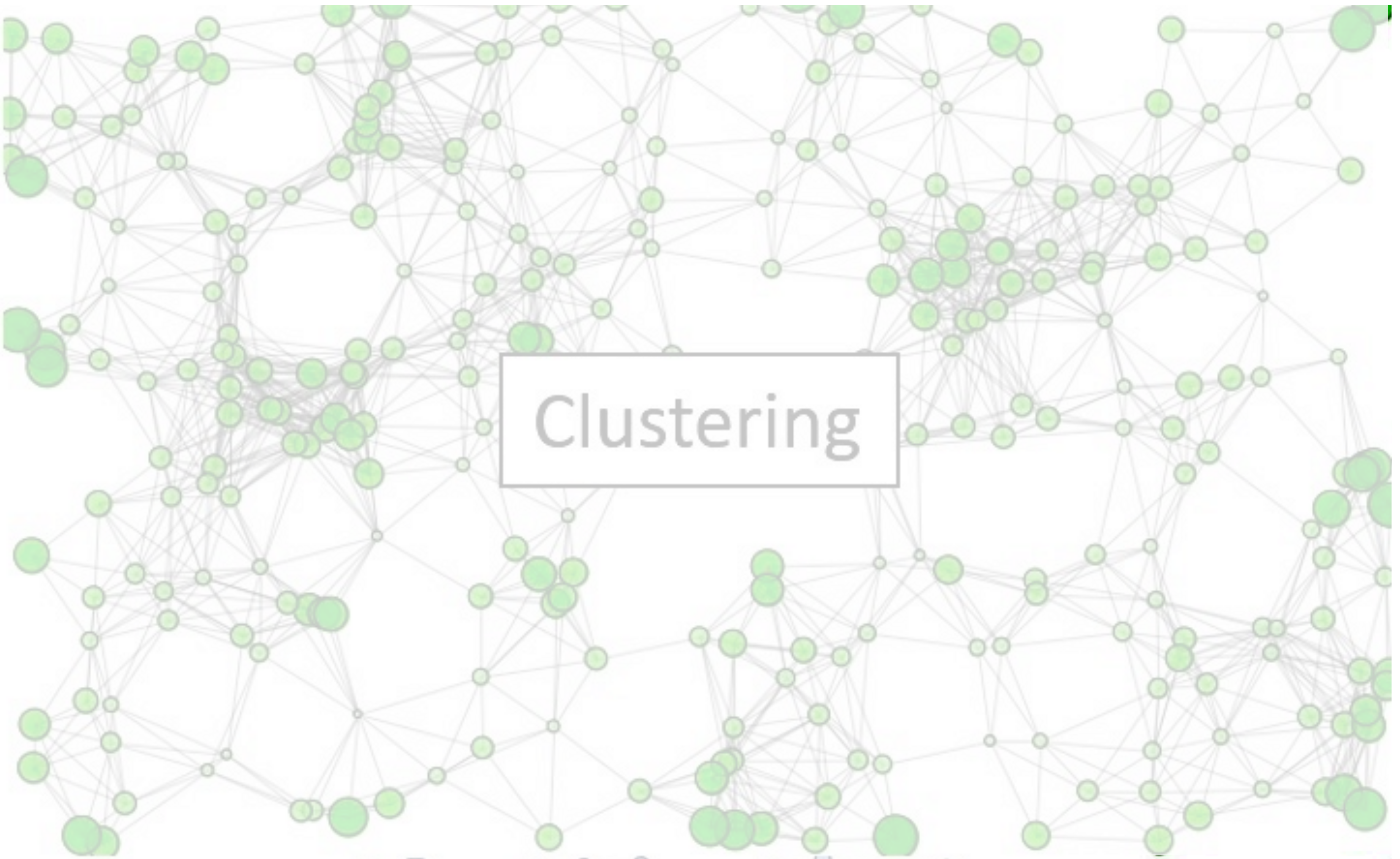
Clustering

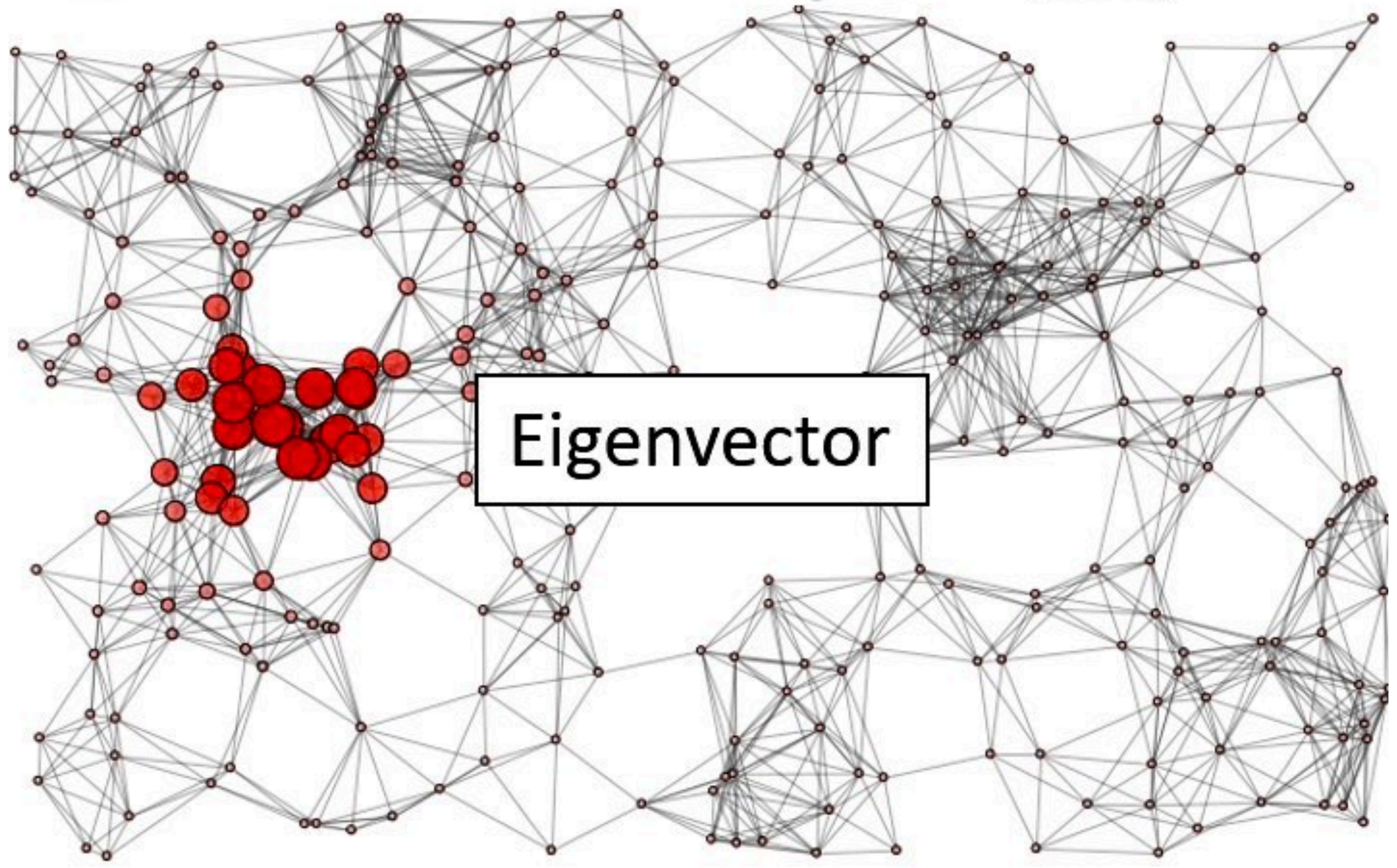


A Visual Wrap-up

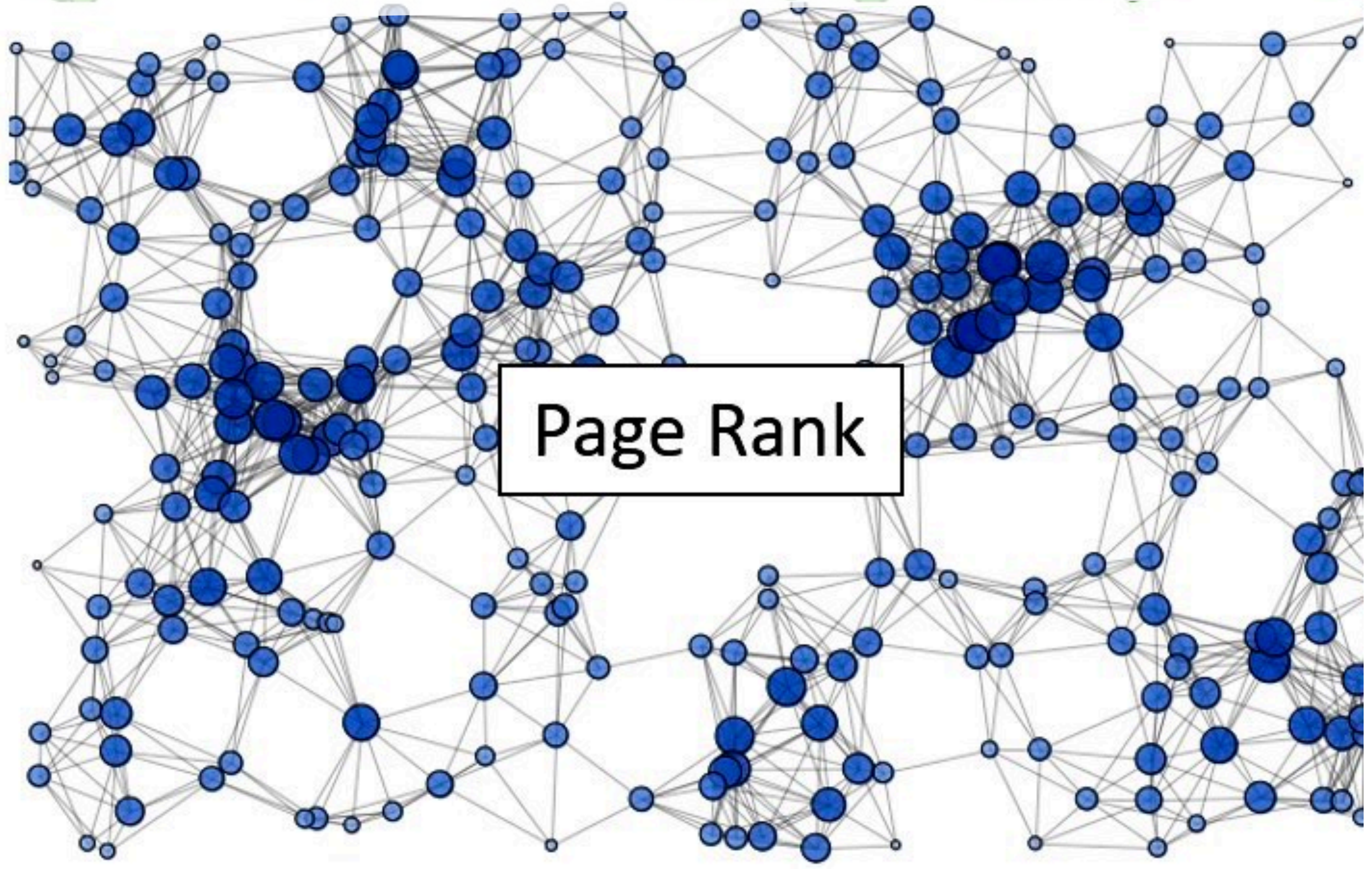




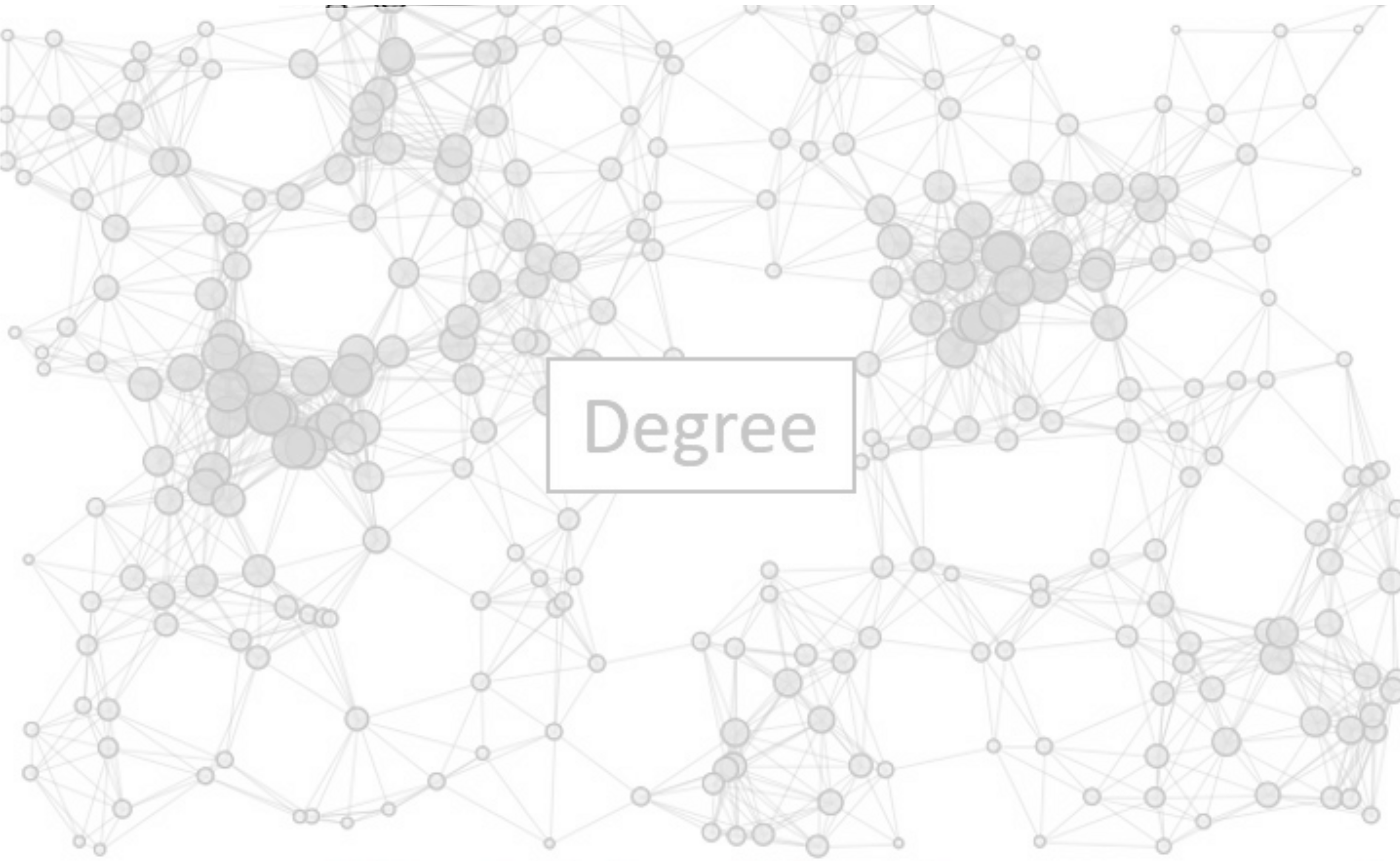


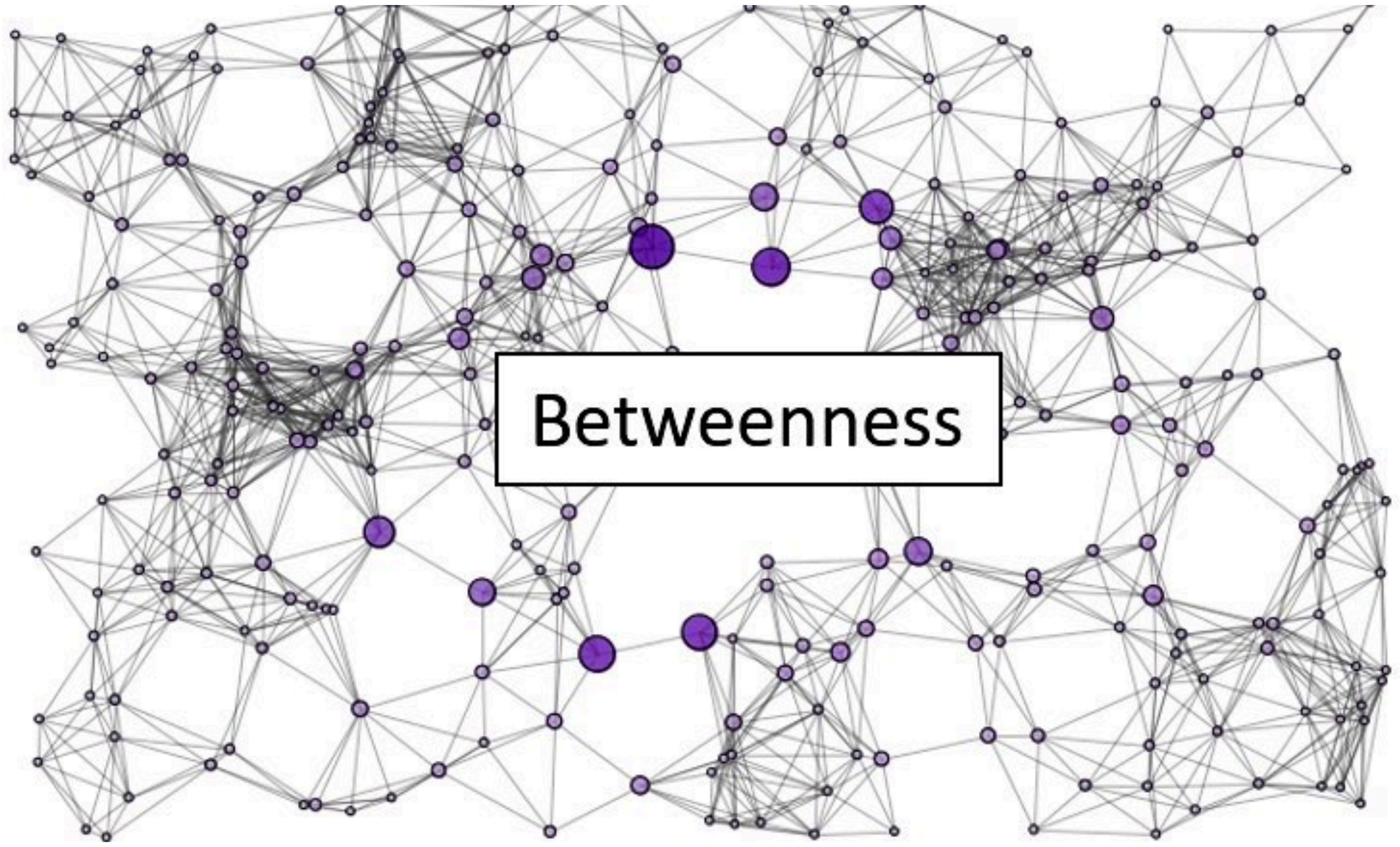


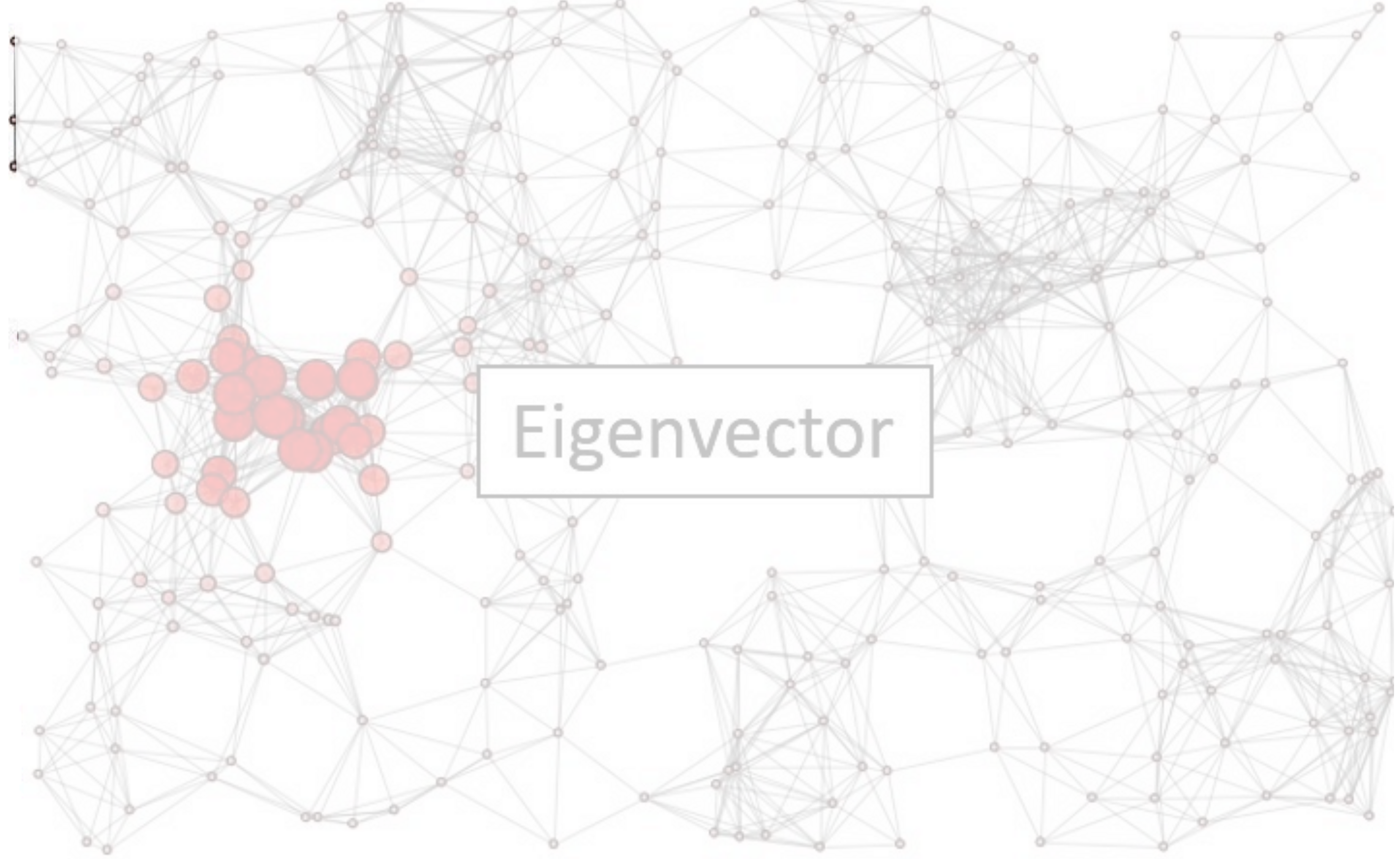
the number of connections and how important are the neighbours of a node



A Visual Wrap-up

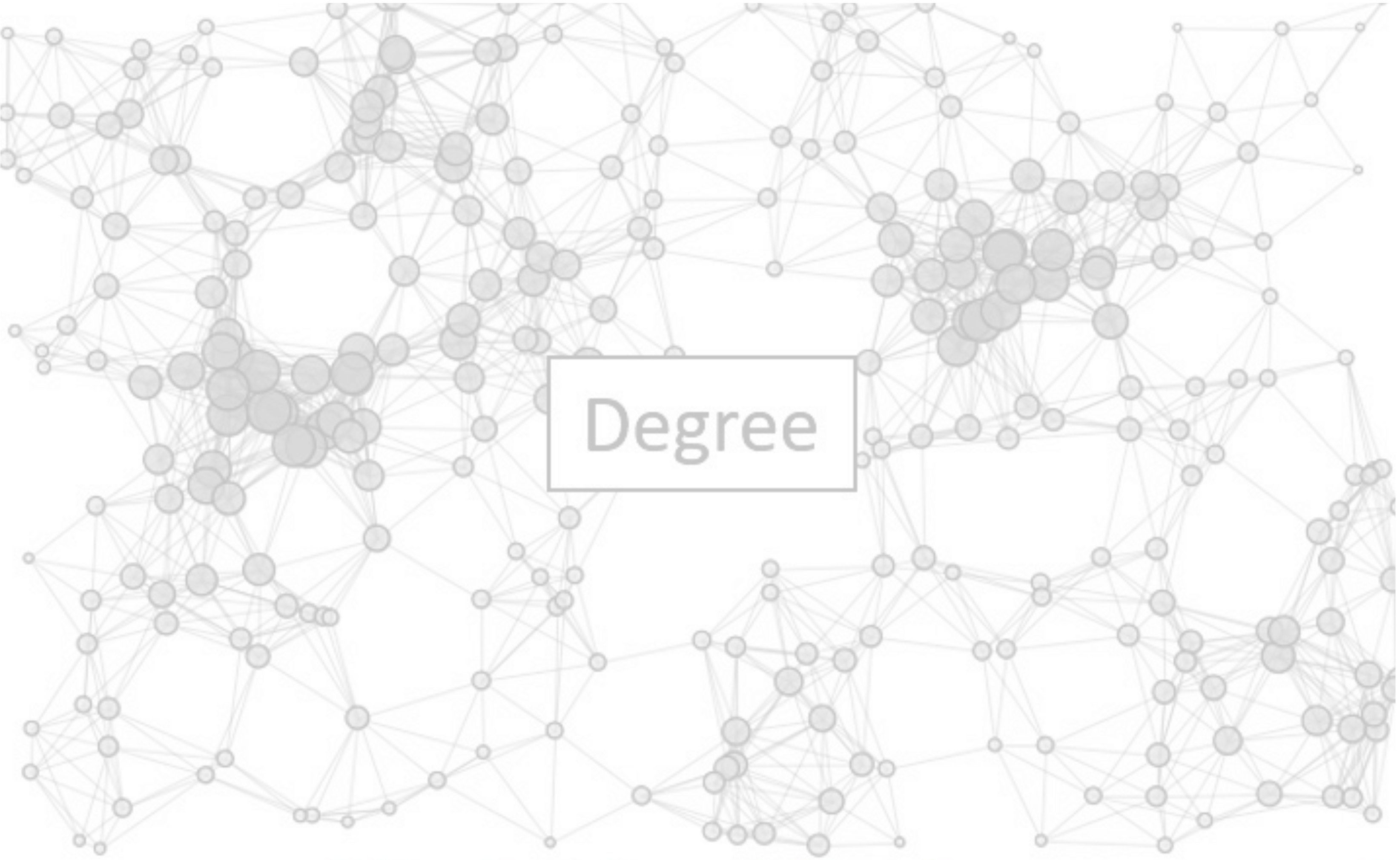






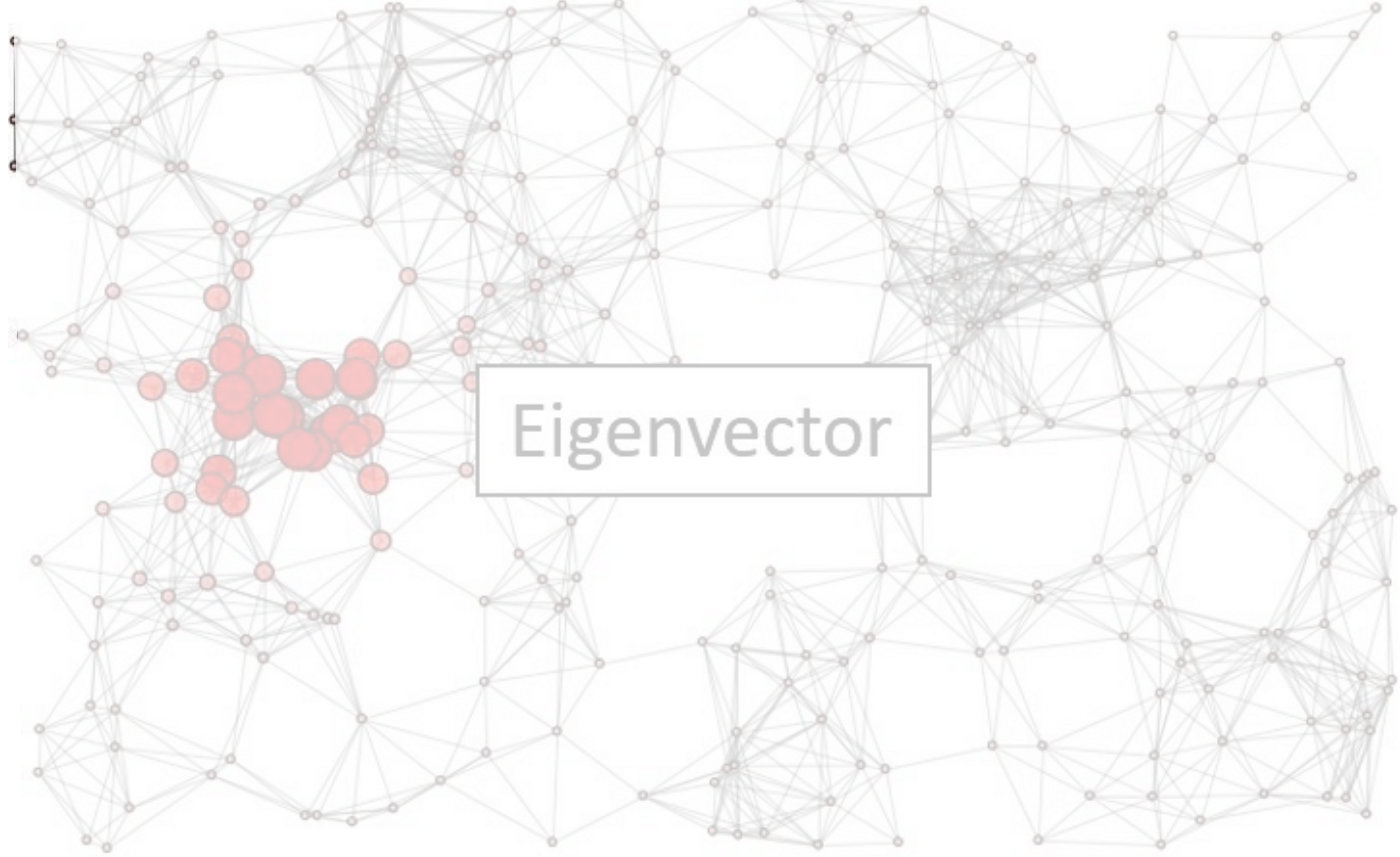
The extent to which a node lies on paths between other nodes

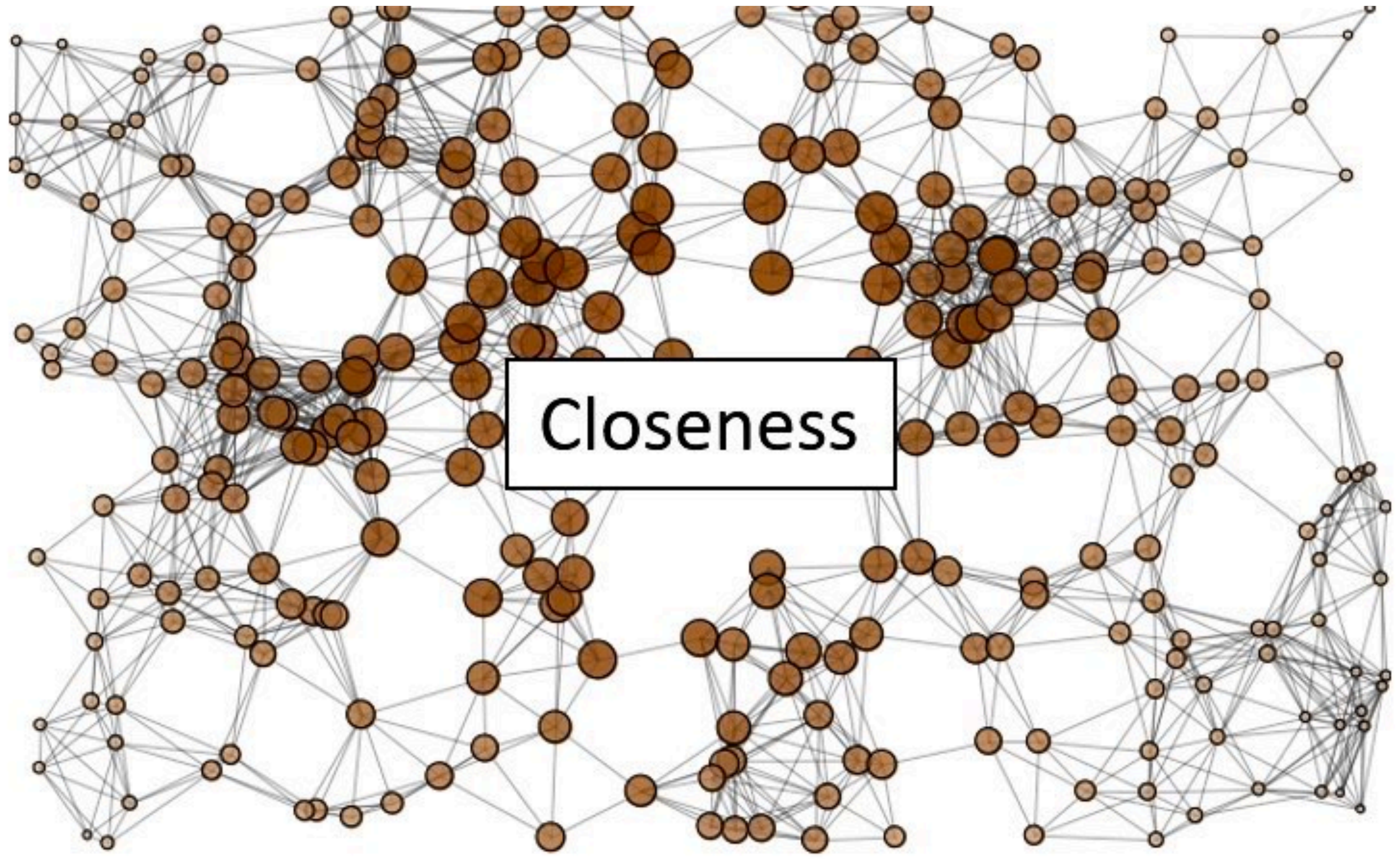
A Visual Wrap-up



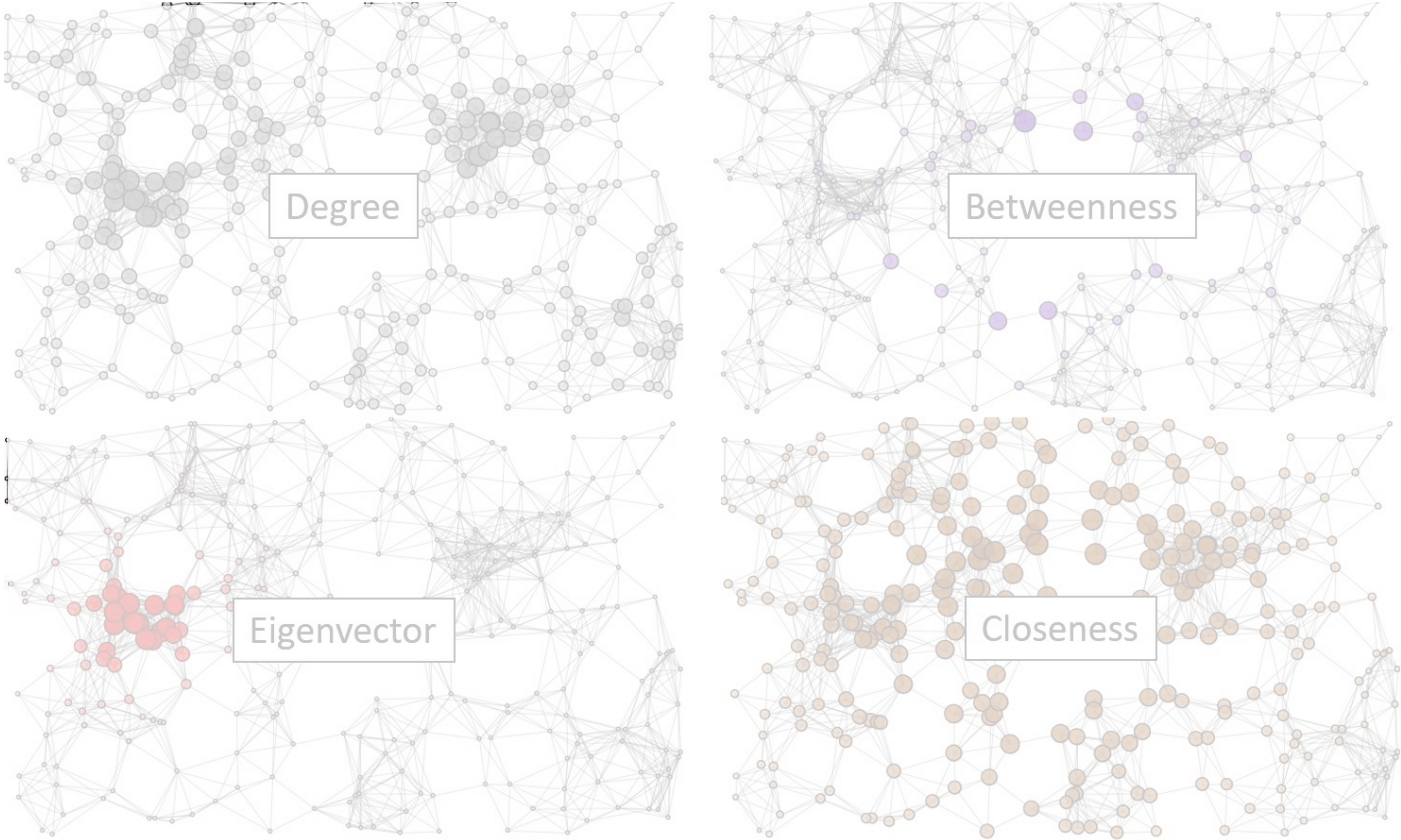
How close a node is to the other nodes

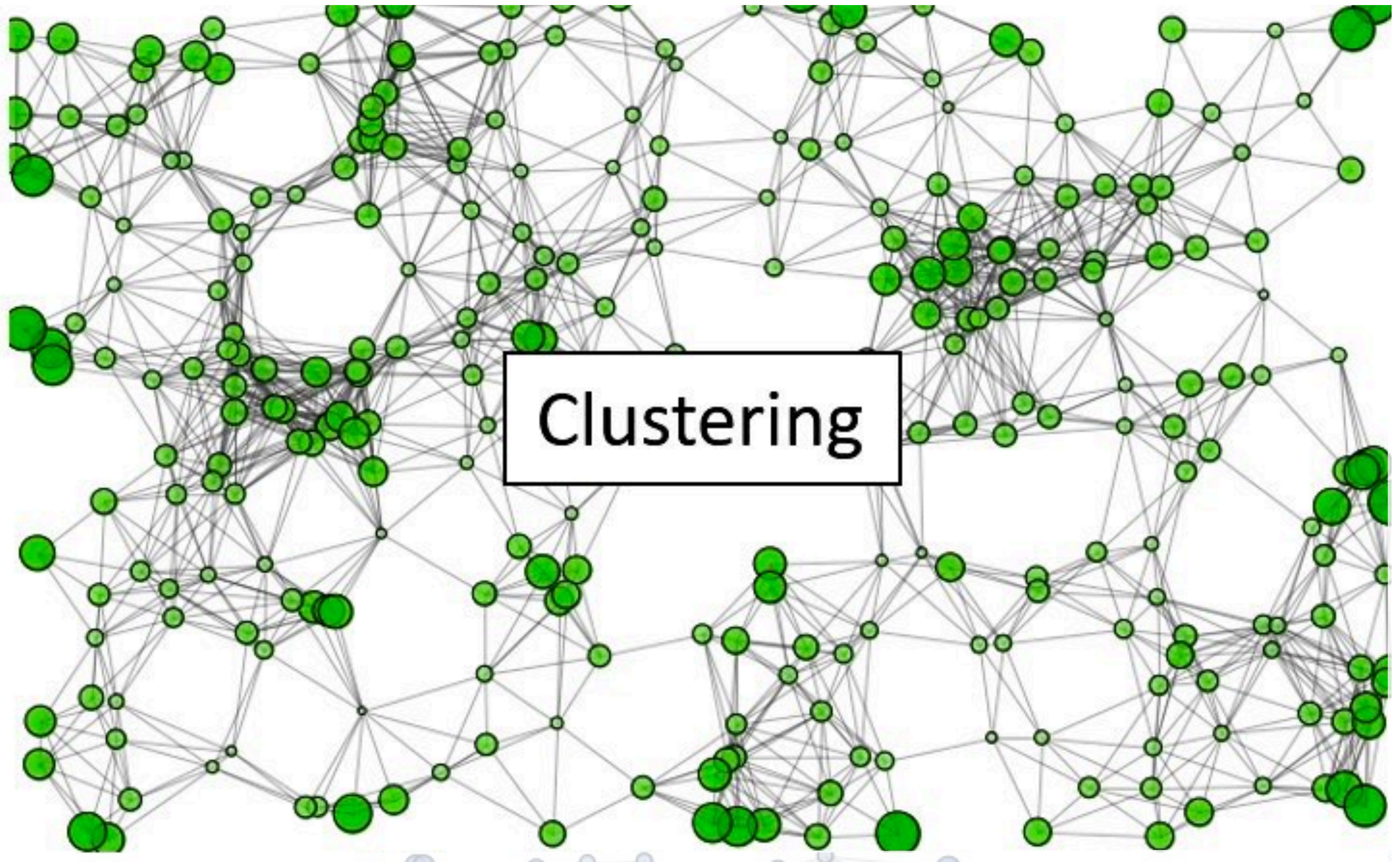
Clustering





A Visual Wrap-up





the likelihood of the neighbours of a node of being neighbours as well

Similarity

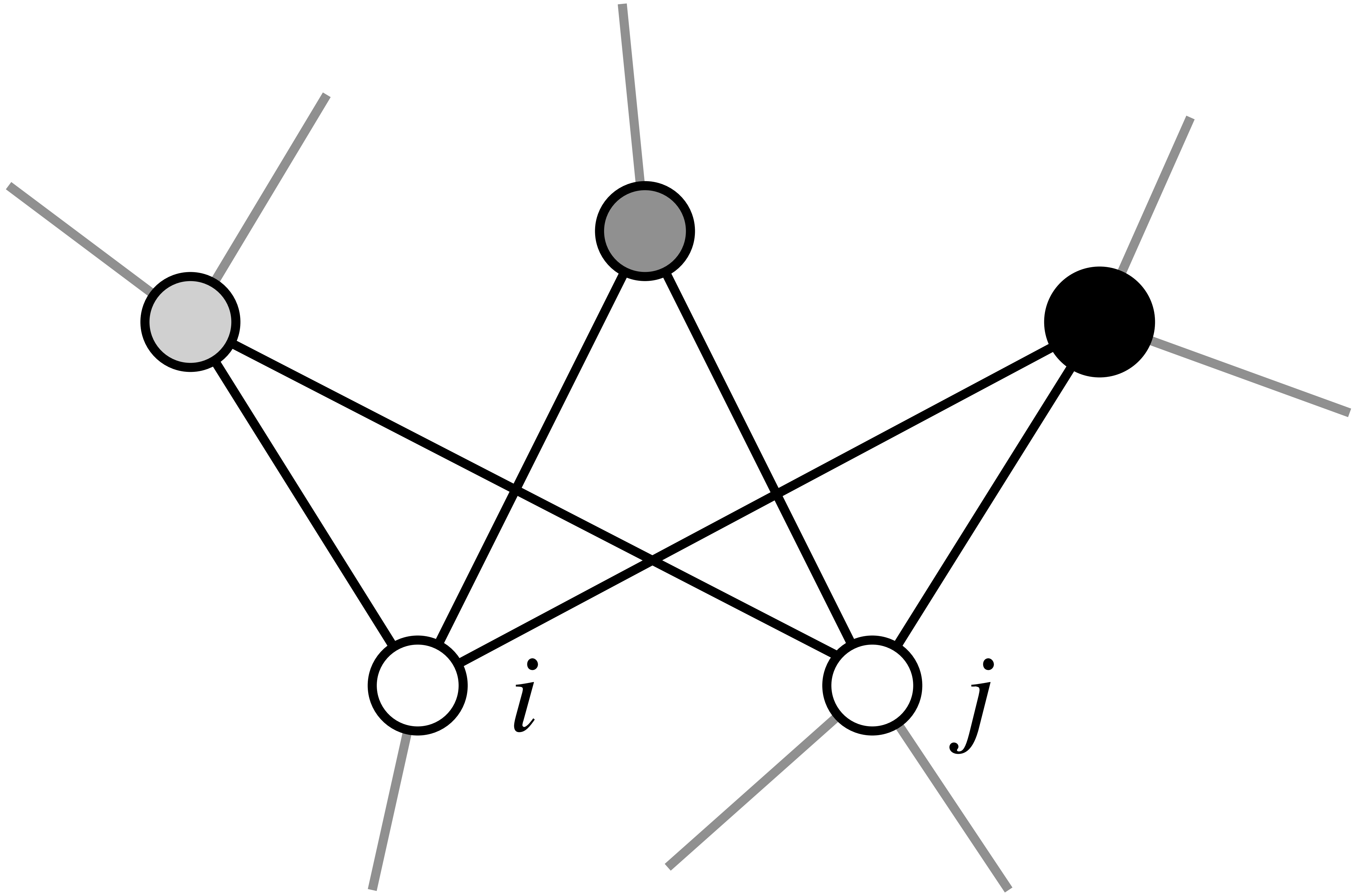
Networks show patterns and repeated node configurations. The most basic repetition is that of nodes that have similar properties. Node similarity can answer to questions like “how unique a node is” and “what are the groups of similar nodes in the network”.

Similarity can abstract from the network, e.g., match-making services match people by similarity using their (self-reported) interests, likes, and dislikes. When looking at similarity from a network perspective, we look at the information contained in the network structure and use that to measure how “distant” are two given nodes in the network.

There are two fundamental measures of network similarity: **structural equivalence** and **regular equivalence**.

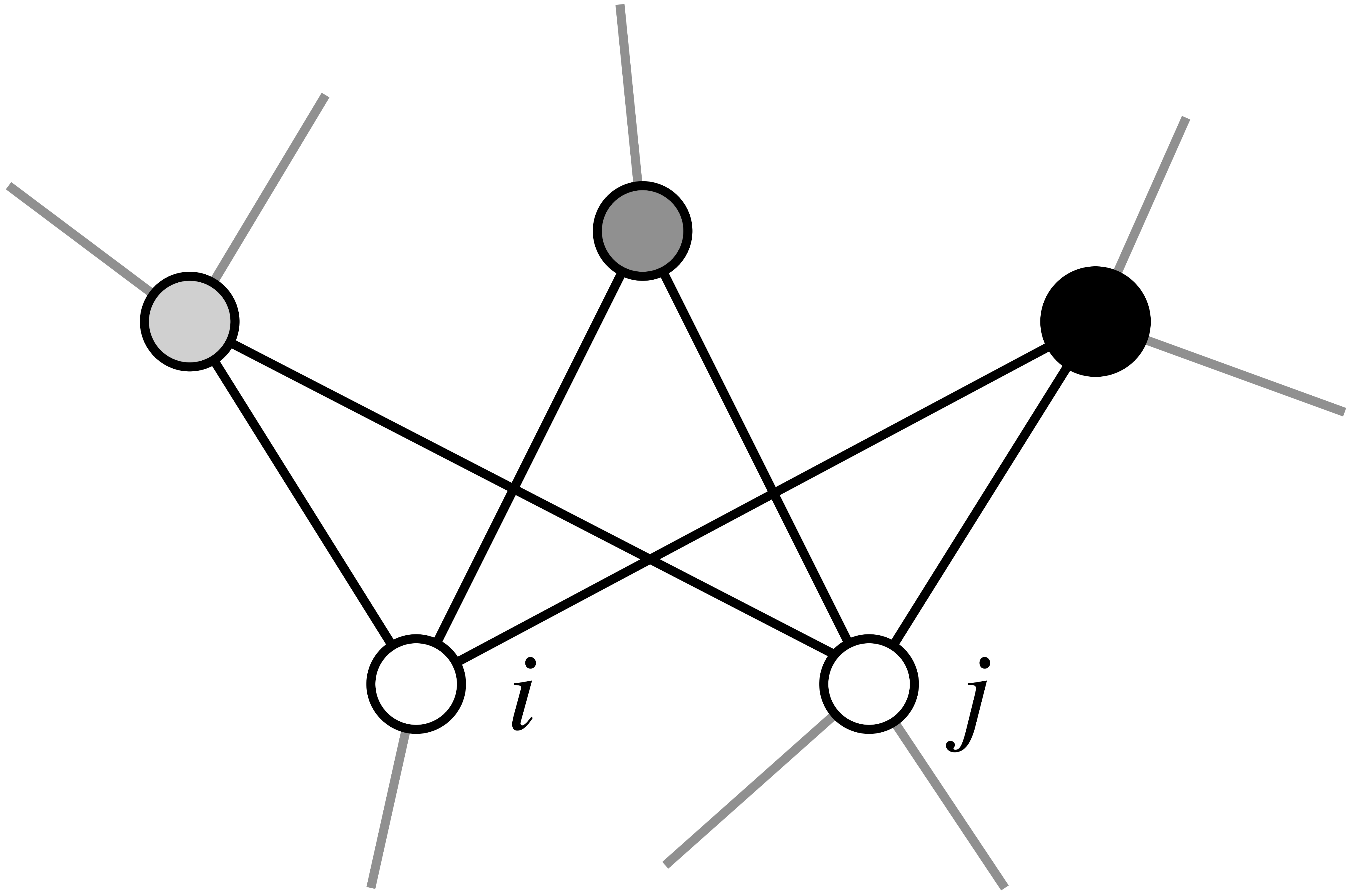
Structural Equivalence

Structural equivalence is a count of the number of common neighbours two nodes have. In an undirected network, the number n_{ij} of common neighbours of nodes *i* and *j* is given by $n_{ij} = \sum_k A_{ik} \, A_{kj}$



Structural Equivalence

However, focussing on the total number of nodes penalises nodes with low degree. The **cosine similarity** is a similarity that compounds varying degrees of nodes.



It is based on a proposal by Salton who suggested to consider the *i*-th and *j*-th rows (colums) of the adjacency matrix as two vectors and use the cosine of the angle θ between them as a measure of their closeness. Formally:

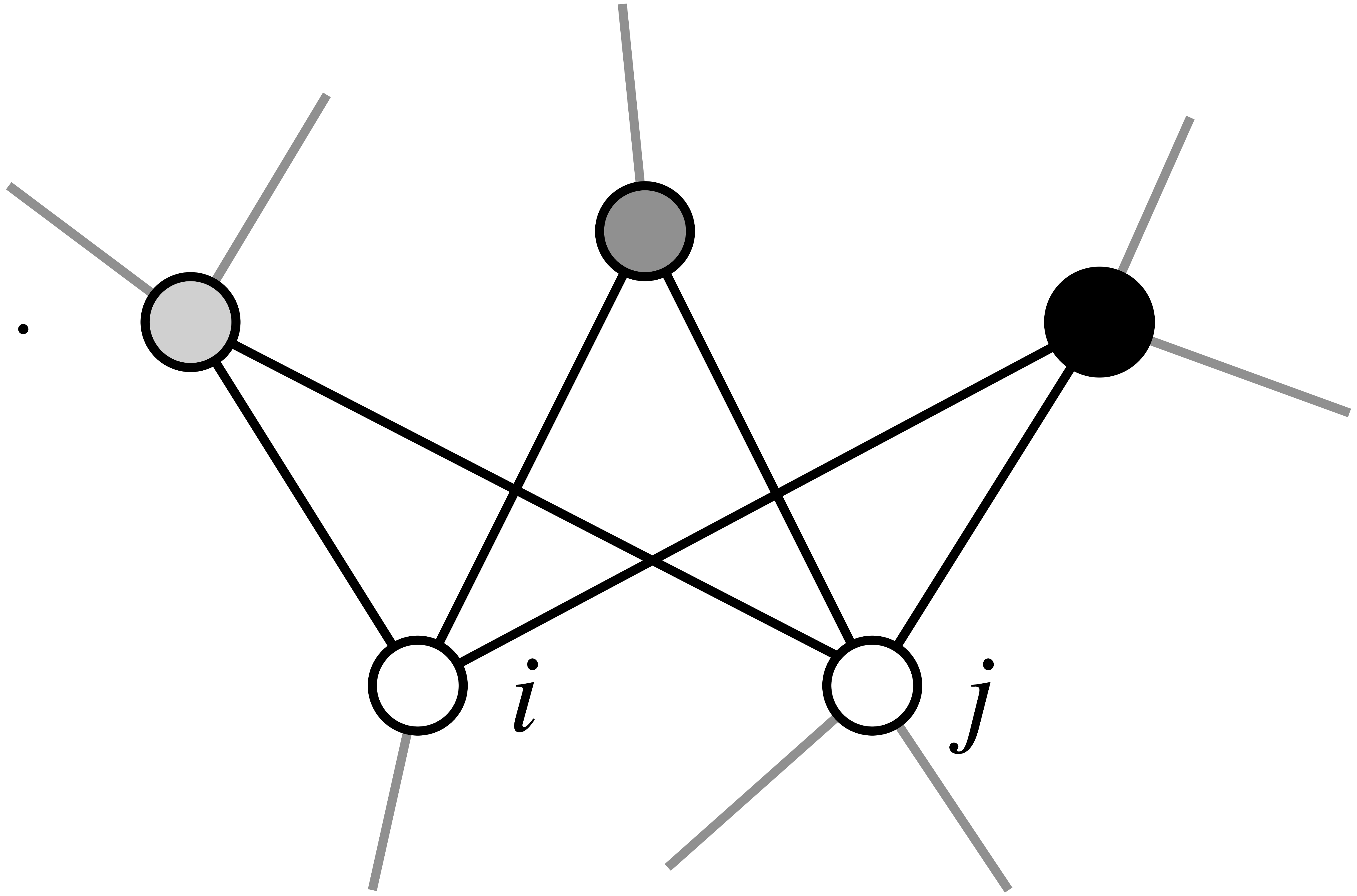
$$\frac{\sum_k A_{ik}A_{kj}}{\sqrt{\sum_k A_{ik}^2}\sqrt{\sum_k A_{jk}^2}} \underset{\text{For unweighted networks}}{=} \frac{\sum_k A_{ik}A_{kj}}{\sqrt{d_i}\sqrt{d_j}} = \frac{n_{ij}}{\sqrt{d_i d_j}}$$

The number of common neighbours of the two nodes

The geometric mean of their degrees

Structural Equivalence

$$\sigma_{ij} = \cos\theta = \frac{n_{ij}}{\sqrt{d_i d_j}} = \frac{3}{\sqrt{4 \times 5}} = 0.671\ldots$$



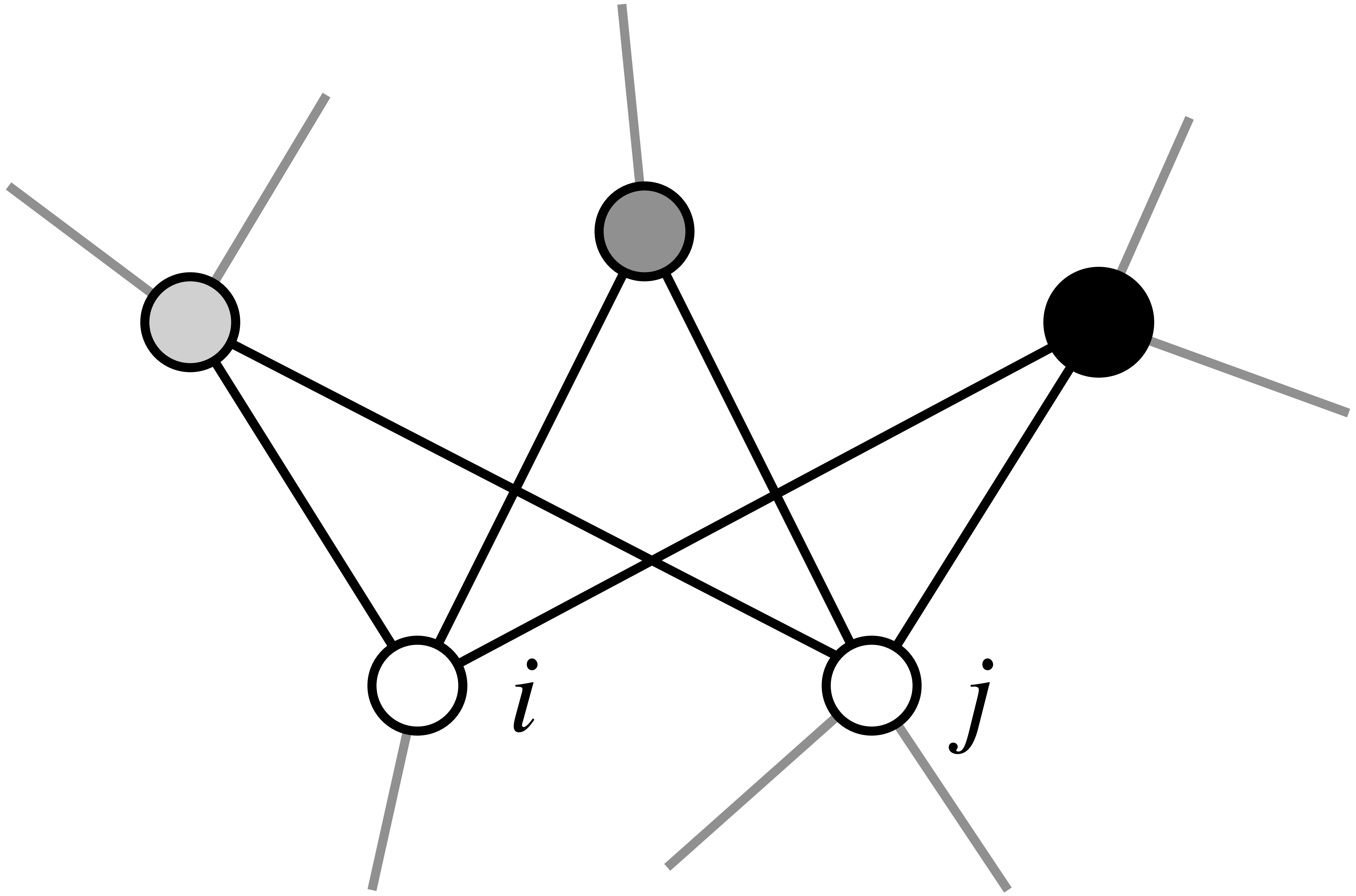
Note that, if the degree of at least one of the nodes is 0, the cosine similarity is undefined, however the convention in those cases is to set it to 0.

Structural Equivalence

There are alternative measures to cosine similarity:

For the **Jaccard coefficient** of two nodes *i* and *j* corresponds to the number of common neighbours n_{ij} divided by the total number of distinct neighbours of both nodes.

$$J_{ij} = \frac{n_{ij}}{d_i + d_j - n_{ij}}$$

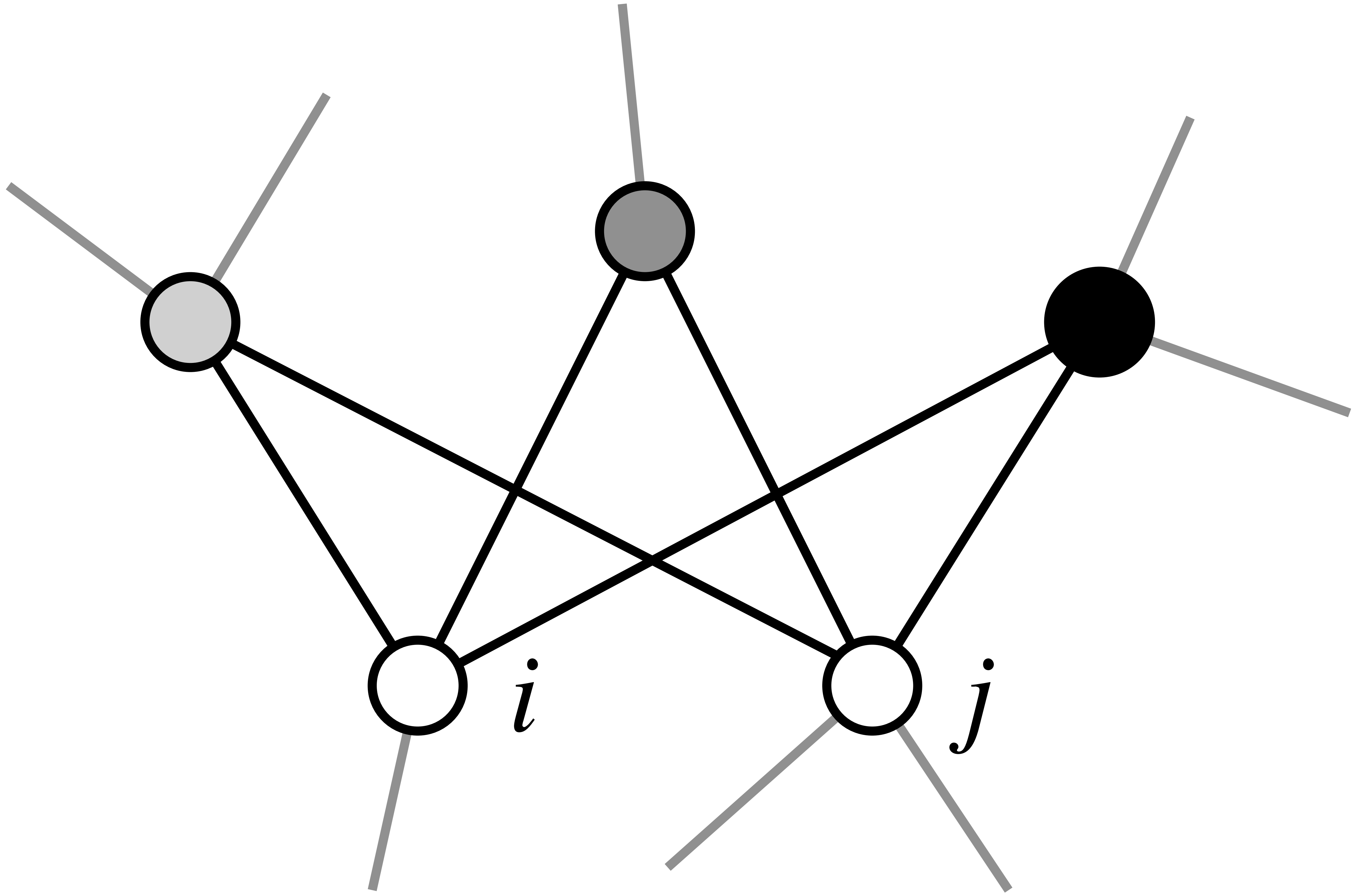


We remove the nodes in common from the sum of the degrees to obtain the amount of distinct nodes

Structural Equivalence

There are alternative measures to cosine similarity:

Pearson correlation coefficient expresses the degree of linear association between two variables. It varies between -1 (antithetical connections), 0 (no correlation), +1 (identity). Pearson is usually applied to scalar or ordinal values.



$$r_{ij} = \frac{\sum_k ((A_{ik} - \langle A_i \rangle) \sum_k ((A_{jk} - \langle A_j \rangle)}{\sqrt{\sum_k (A_{ik} - \langle A_i \rangle)^2} \sqrt{\sum_k (A_{jk} - \langle A_j \rangle)^2}}$$

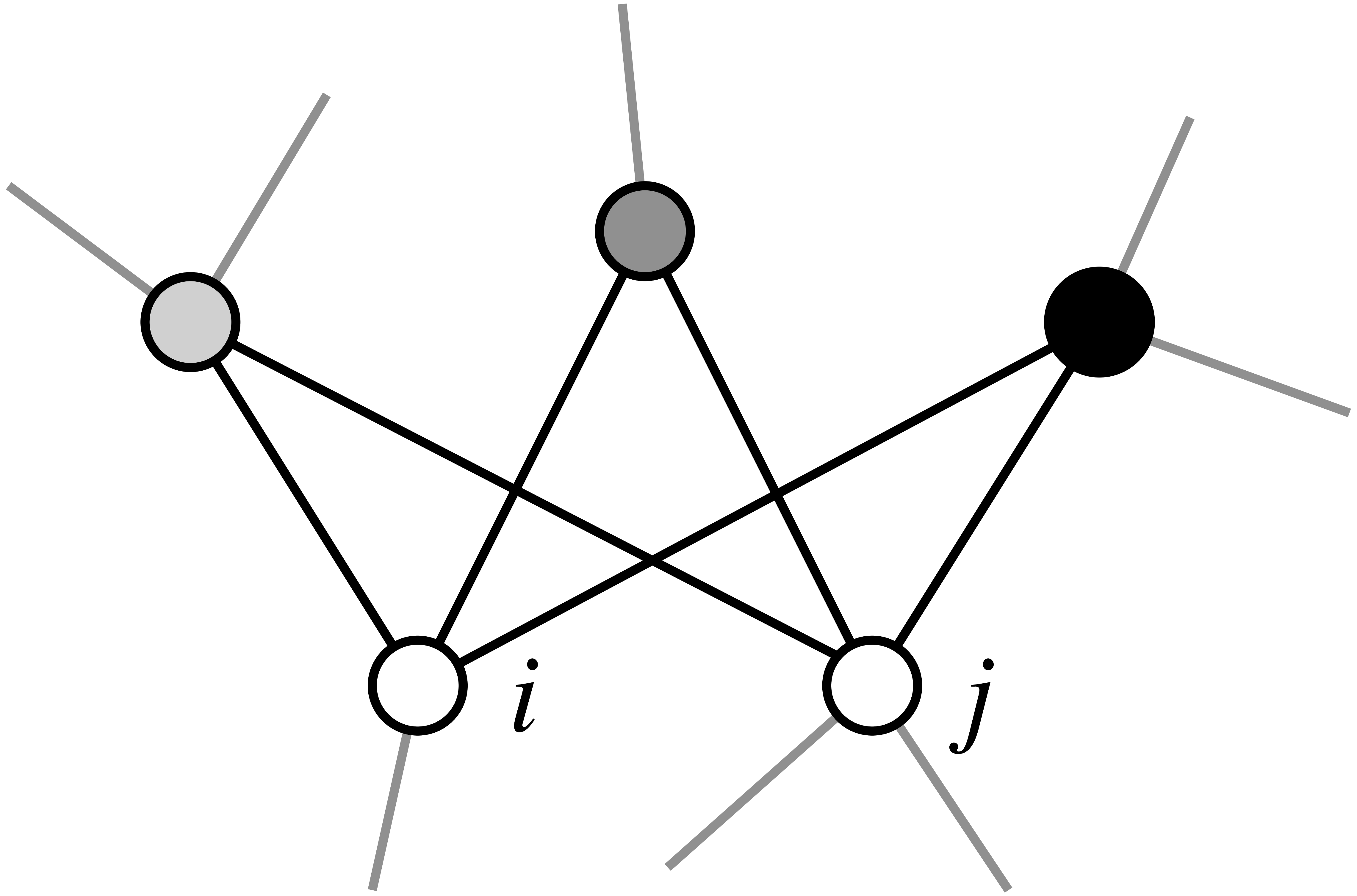
Average of the *-th row

Structural Equivalence

There are alternative measures to cosine similarity:

Hamming distance which calculates the number of neighbours two nodes do not have in common. It can be interpreted also as the number of ties a node *i* must change to take the place of a node *j* - the square root of h_{ij} is called the **Euclidean distance** between *i* and *j*.

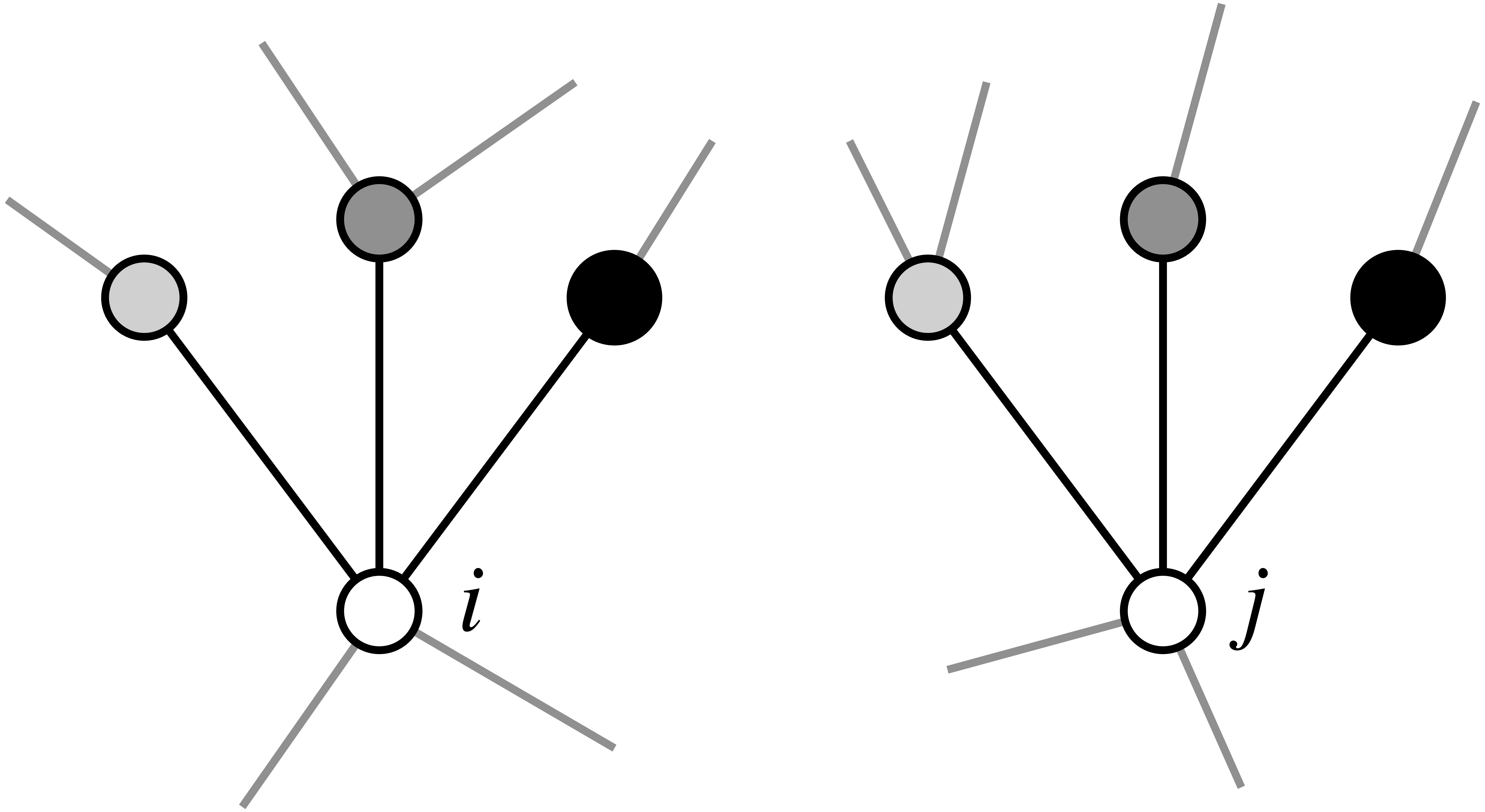
$$h_{ij} = \sum_k (A_{ik} - A_{jk})^2$$



Regular Equivalence

Regular equivalence of two nodes is the count of neighbours that *are themselves similar*.

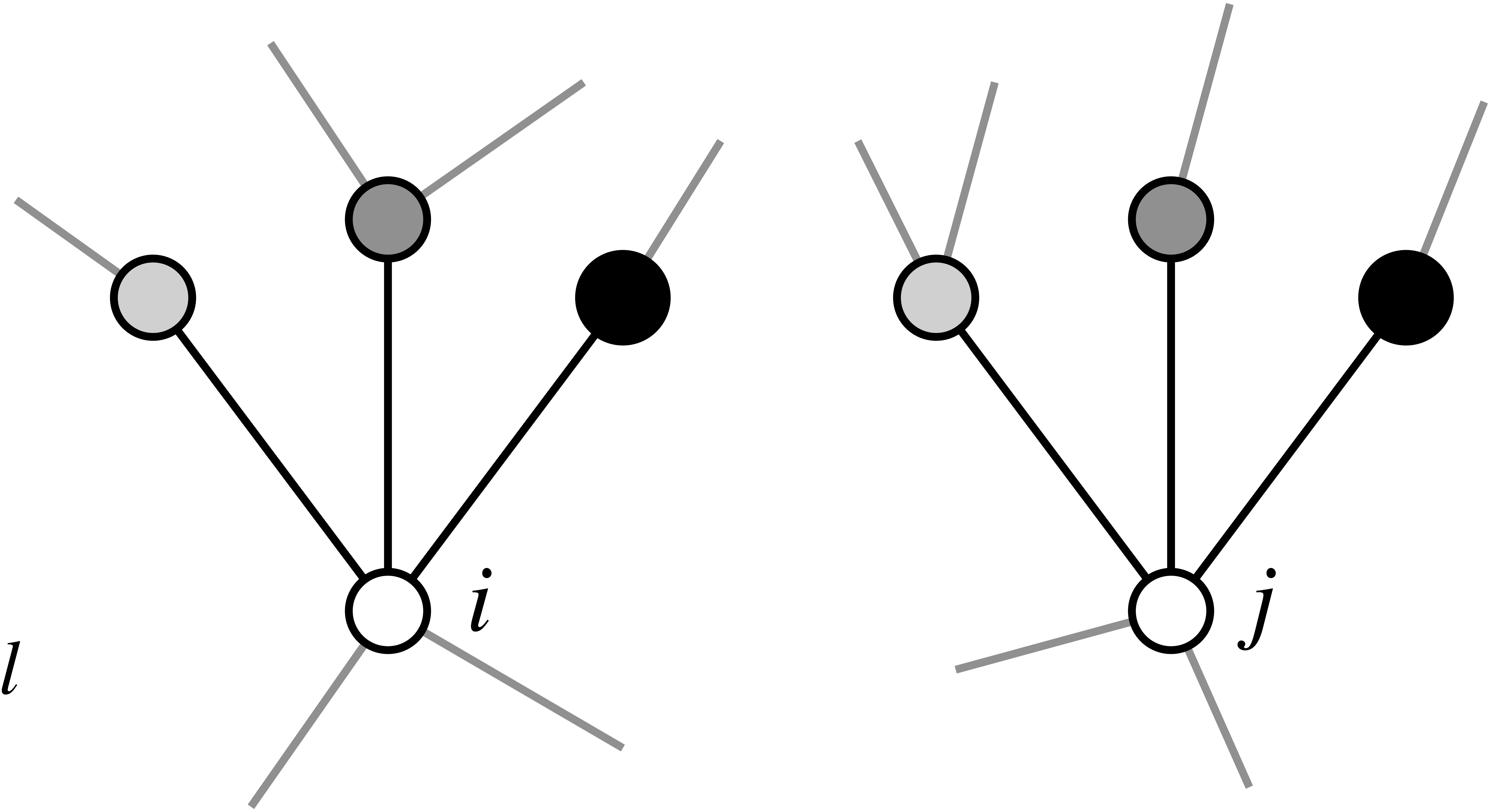
E.g., two CEOs at different companies may have no colleagues in common, but they are similar in the sense that they have professional ties to their respective CFO, CIO, members of the board, company president, and so forth.



Regular Equivalence

The basic idea is to define a similarity score σ_{ij} such that *i* and *j* have high similarity if they have neighbours *k* and *l* that themselves have high similarity. For an undirected network we have

$$\sigma_{ij} = \alpha \sum_{kl} A_{ik}\, A_{jl}\, \sigma_{kl}$$

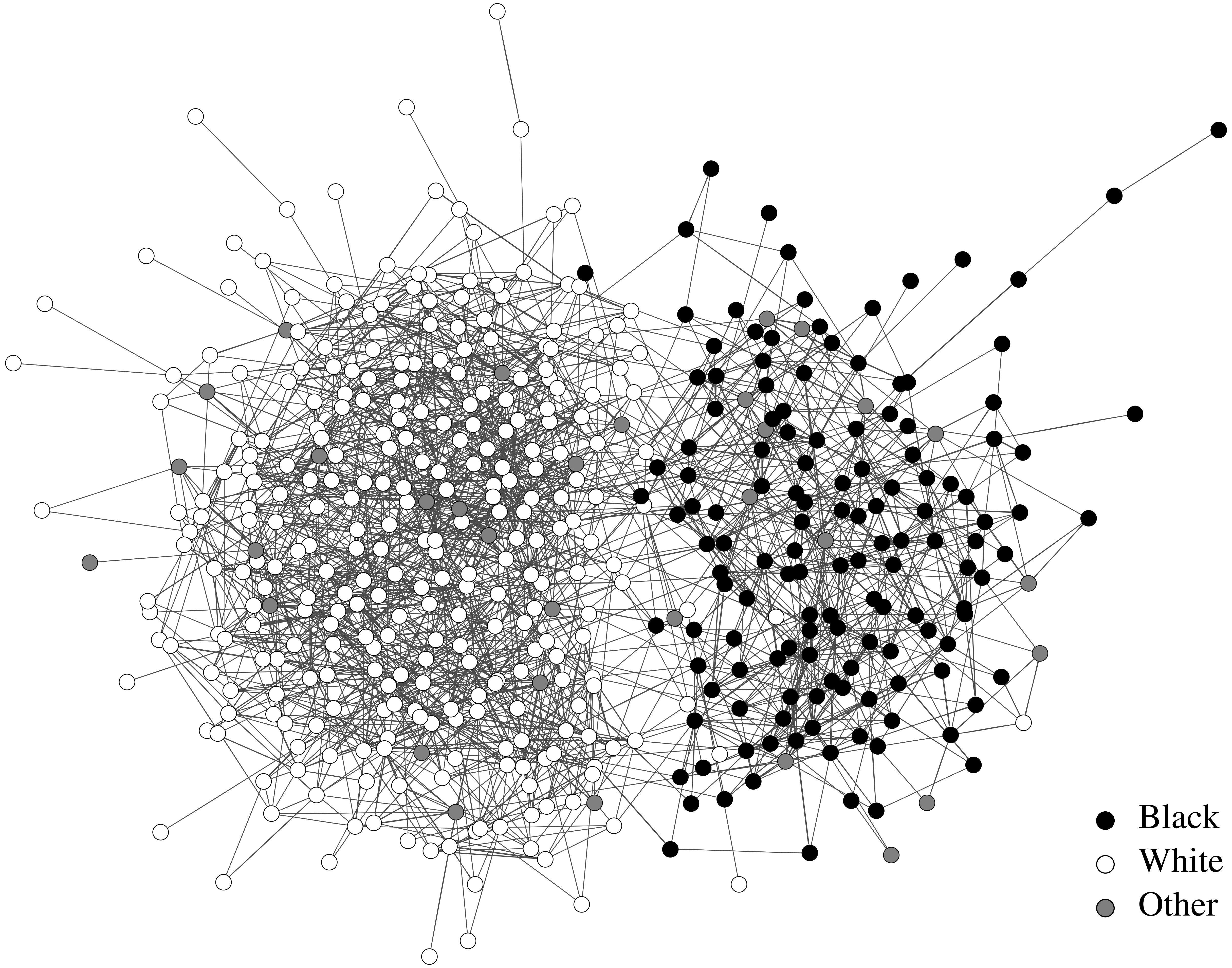


With α constant (the inverse of the eigenvalue) and σ the leading eigenvector … with all the considerations we discussed for centrality, Kats measure, PageRank …

Homophily and Assortative Mixing

Related to similarity and equivalence, *homophily* (also called *assortative mixing*) reports the tendency of nodes in the network to draw ties with other nodes that are similar/equivalent to them.

For example, a large body of literature shows how ethnic segregation does not strictly relate to an extremist aversion against other ethnicities (e.g., 90/10 ratio), but it can also emerge from moderate preference ratios (e.g., 55/45) of same vs. other ethnic groups.



Homophily and Assortative Mixing • by Unordered Characteristics

A network is assortative if a significant fraction of the edges in the network run between nodes of the same type.

To measure the level of assortativity, we can calculate a) *the fraction of edges that run between nodes of the same type and subtract from that figure b) the fraction of such edges we would expect to find if edges were positioned at random without regard for node type*. Hence, this measure is in a sense quantifying the level of “non-randomness” in the placement of edges in the network.

First, we calculate a)

Kronecker delta, where $\delta_{kl} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{otherwise} \end{cases}$

$$a) = \frac{1}{2} \sum_{ij} A_{ij} \delta_{g_i g_j}$$

The group/class/type of node *i*; g_i is an integer $1 \quad \dots \quad N$ where *N* is the total number of groups

Homophily and Assortative Mixing • by Unordered Characteristics

A network is assortative if a significant fraction of the edges in the network run between nodes of the same type.

To measure the level of assortativity, we can calculate a) *the fraction of edges that run between nodes of the same type and subtract from that figure b) the fraction of such edges we would expect to find if edges were positioned at random without regard for node type*. Hence, this measure is in a sense quantifying the level of “non-randomness” in the placement of edges in the network.

And then we calculate b)

There can be 2*m* ends of edges in the entire network (with *m* being the number of edges). Given an edge with an end at *i*, the chance that the other end belongs to *j* is $d_j/2m$.

$$b) = \frac{1}{2}\sum_{ij}\frac{d_j}{2m}\,d_i\,\delta_{g_i\,g_j}$$

We repeat that measure for all edges ending in *i*

Same measure for the “actual” nodes used in a)

Homophily and Assortative Mixing • by Unordered Characteristics

A network is assortative if a significant fraction of the edges in the network run between nodes of the same type.

To measure the level of assortativity, we can calculate a) *the fraction of edges that run between nodes of the same type and subtract from that figure b) the fraction of such edges we would expect to find if edges were positioned at random without regard for node type*. Hence, this measure is in a sense quantifying the level of "non-randomness" in the placement of edges in the network.

Putting together a) and b) we have

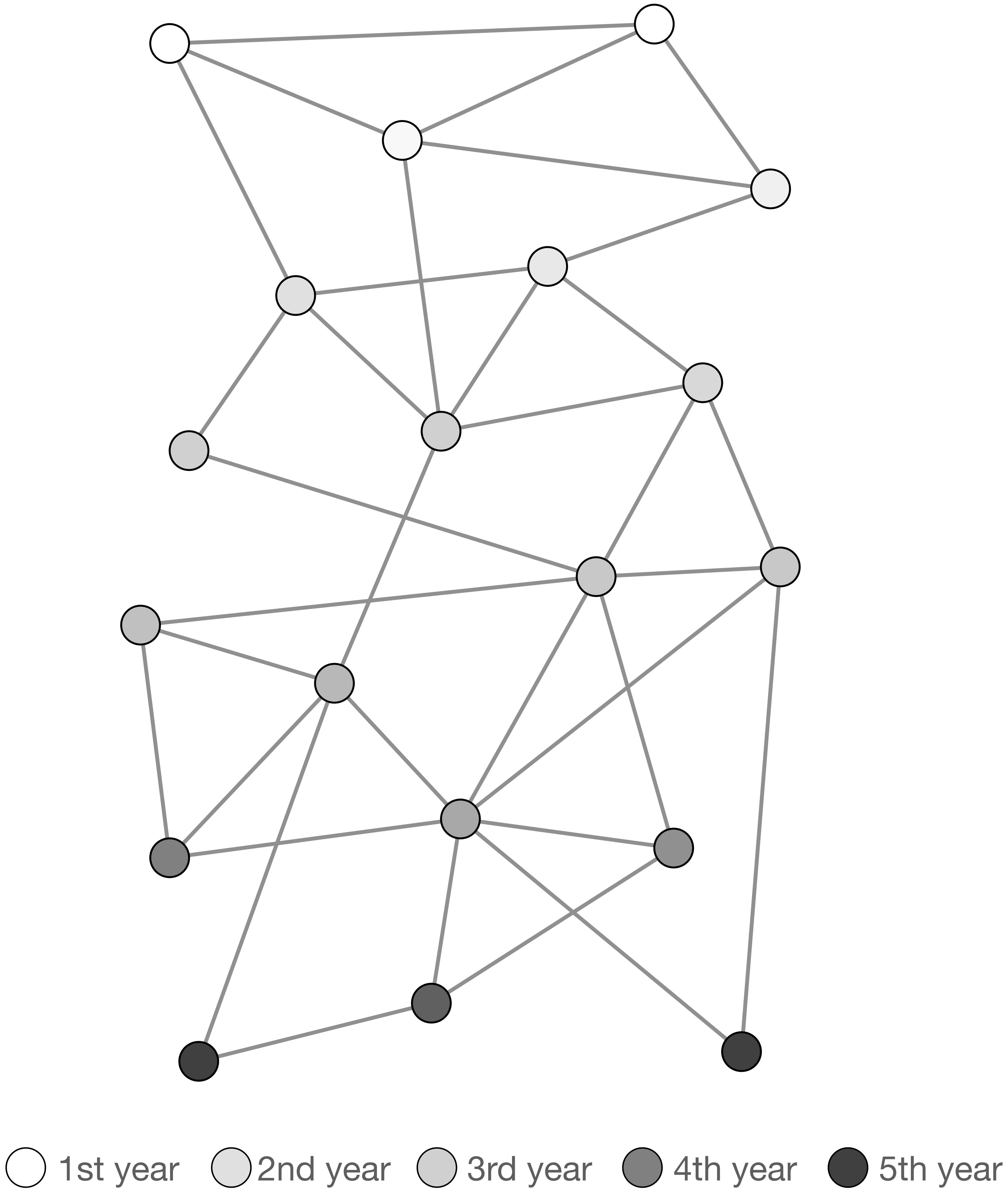
$$a) - b) = \frac{1}{2} \sum_{ij} \left(A_{ij} - \frac{d_j \, d_i}{2m} \right) \delta_{g_i \, g_j}$$

Homophily and Assortative Mixing • by Ordered Characteristics

We can calculate assortative mixing also on networks with ordered characteristics, like age or income, which supports the calculation of approximation of assortativity based on the distance between those characteristics.

If network nodes with similar values of a scalar characteristic tend to be connected together more likely than those with different values, then the network is considered assortatively mixed according to that characteristic.

For example, if people are friends with others around the same age as them, then the network is assortatively mixed (or *stratified*) by age.



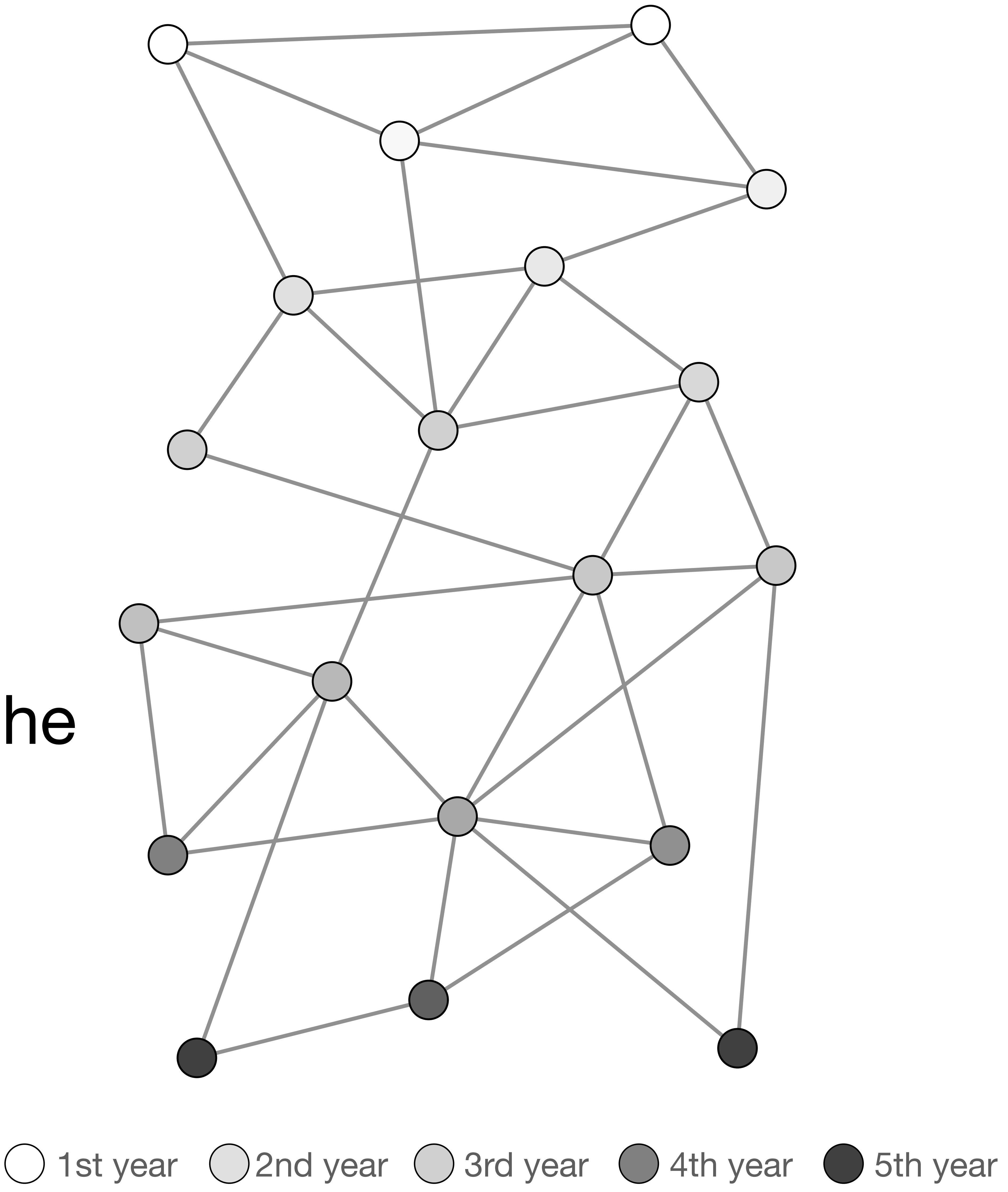
Homophily and Assortative Mixing • by Ordered Characteristics

To measure assortativity on ordered characteristics, we can calculate the covariance of the network. Let us have x_i being the value of attribute x for node i, we have

$$\mathrm{COV}(x_i, x_j) = \frac{\sum_{ij} A_{ij}(x_i - \mu^x)(x_j - \mu^x)}{\sum_{ij} A_{ij}}$$

mean of the value x

$$\mu^x = \frac{\sum_{ij} A_{ij}\ x_i}{\sum_{ij} A_{ij}} = \frac{1}{2m} \sum_i d_i\ x_i$$

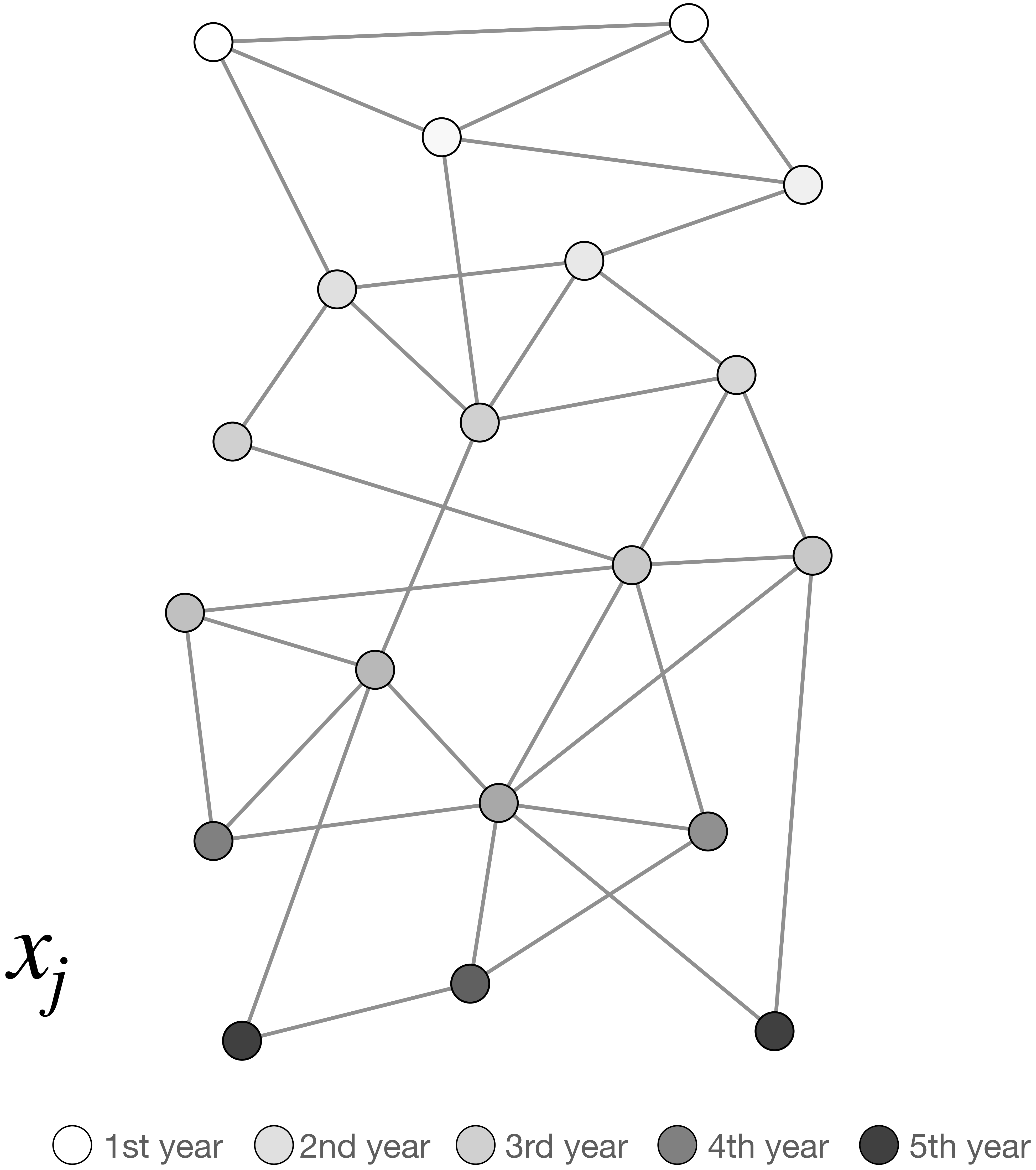


Homophily and Assortative Mixing • by Ordered Characteristics

To measure assortativity on ordered characteristics, we can calculate the covariance of the network. Let us have x_i being the value of attribute *x* for node *i*, we have

$$\mu^x = \frac{1}{2m} \sum_i d_i \; x_i$$

$$\text{COV}(x_i, x_j) = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{d_i \; d_j}{2m} \right) x_i \; x_j$$



Homophily and Assortative Mixing • by Ordered Characteristics

$$\mathrm{COV}(x_i, x_j) = \underbrace{\frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{d_i\, d_j}{2m}\right)}_{}\, x_i\, x_j$$

Assortativity wrt the total number of edges is called **modularity**, denoted Q, and it measures the extent to which similar nodes are likely to connect to each other.

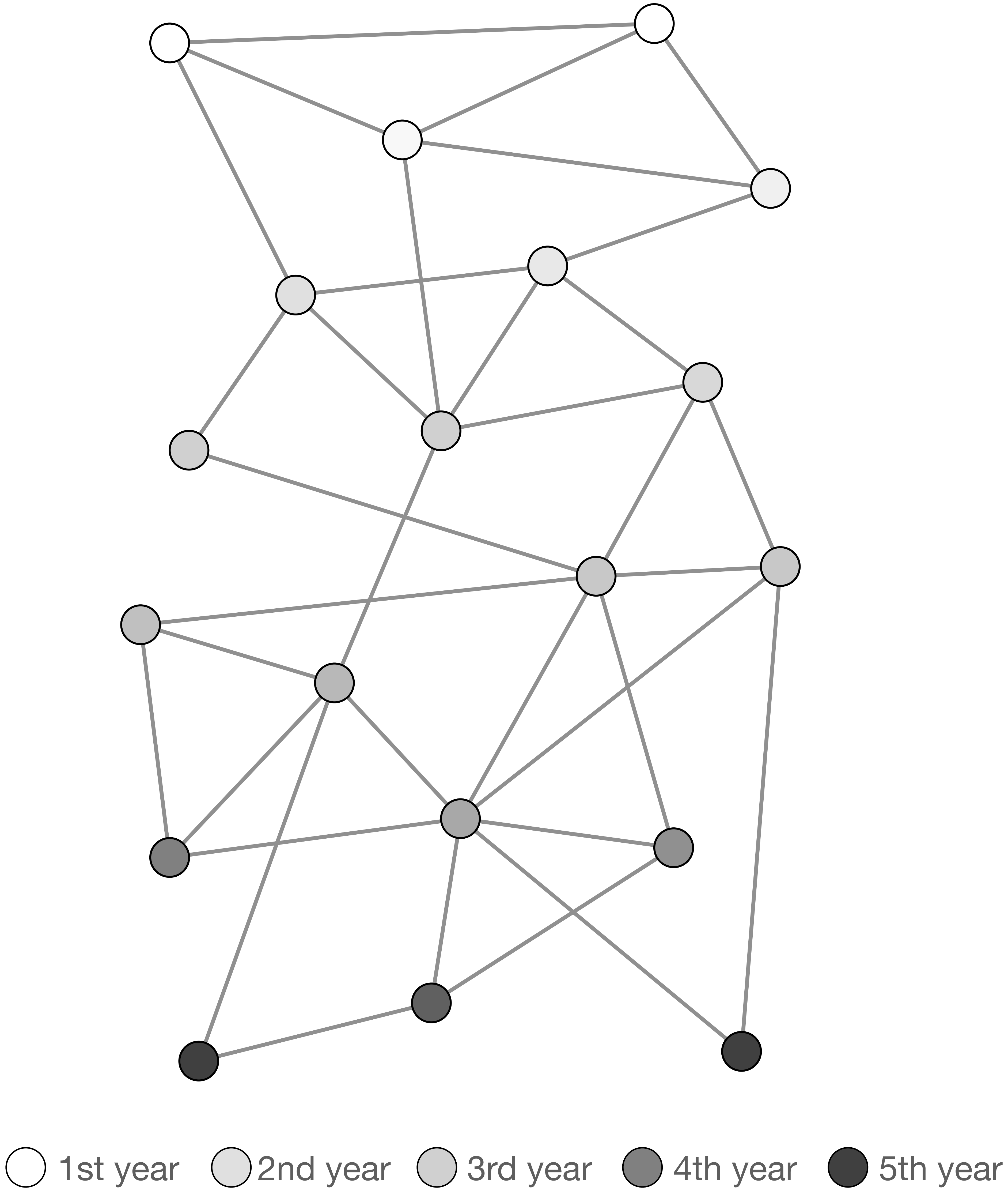
$$Q = \boxed{\frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{d_j\, d_i}{2m}\right)}\,\delta_{g_i\, g_j}$$

Homophily and Assortative Mixing • by Ordered Characteristics

It is sometimes convenient to normalize the covariance so that it takes the value 1 in a network with perfect assortative mixing—one in which all edges fall between nodes with precisely equal values of x_i.

$$r = \frac{\sum_{ij}\left(A_{ij} - \frac{d_i\, d_j}{2m}\right) x_i\; x_j}{\sum_{ij}\left(d_i\delta_{ij} - \frac{d_i\, d_j}{2m}\right) x_i\; x_j}$$

The obtained measure is called the “assortativity coefficient”.

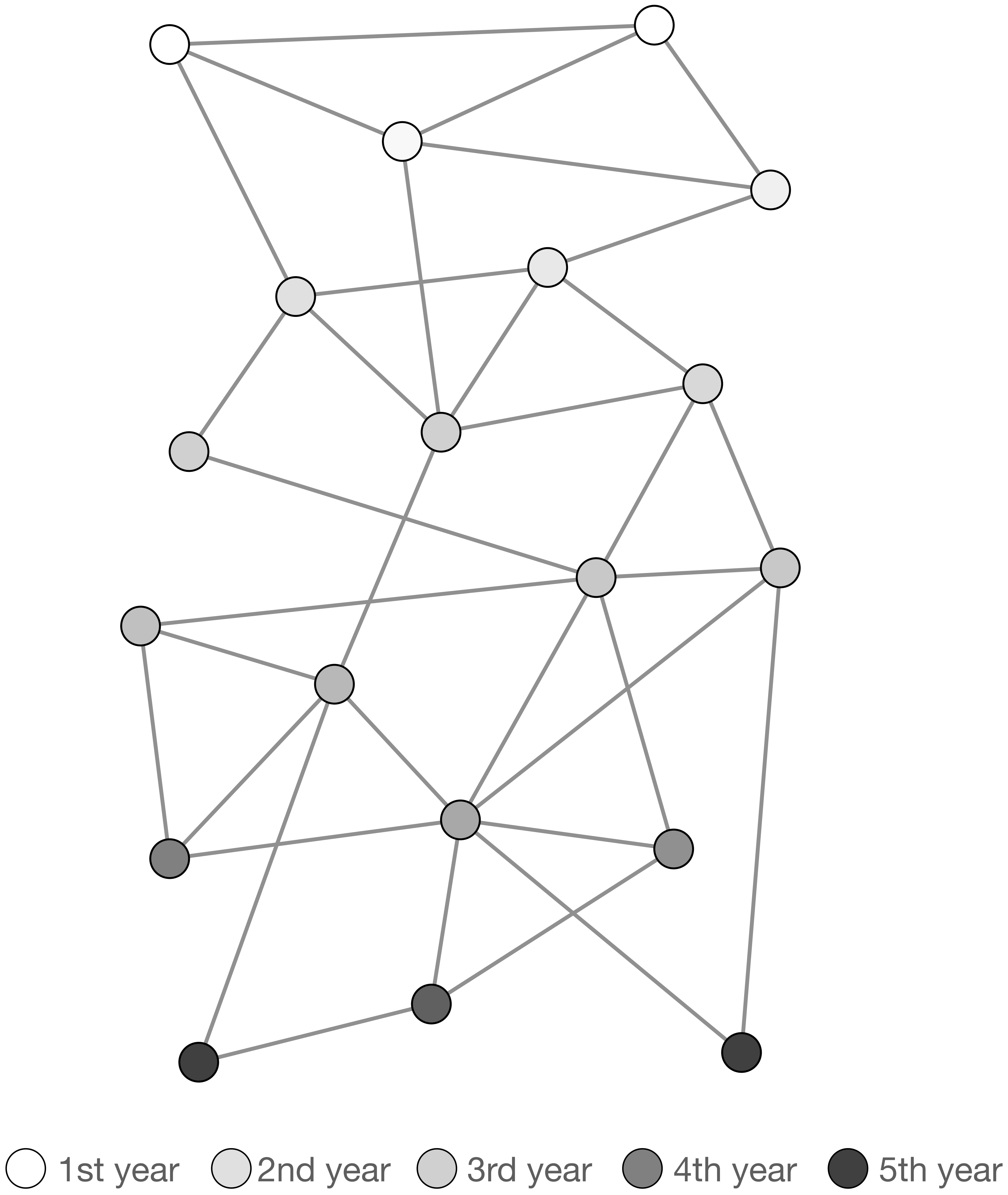


Homophily and Assortative Mixing • by Ordered Characteristics

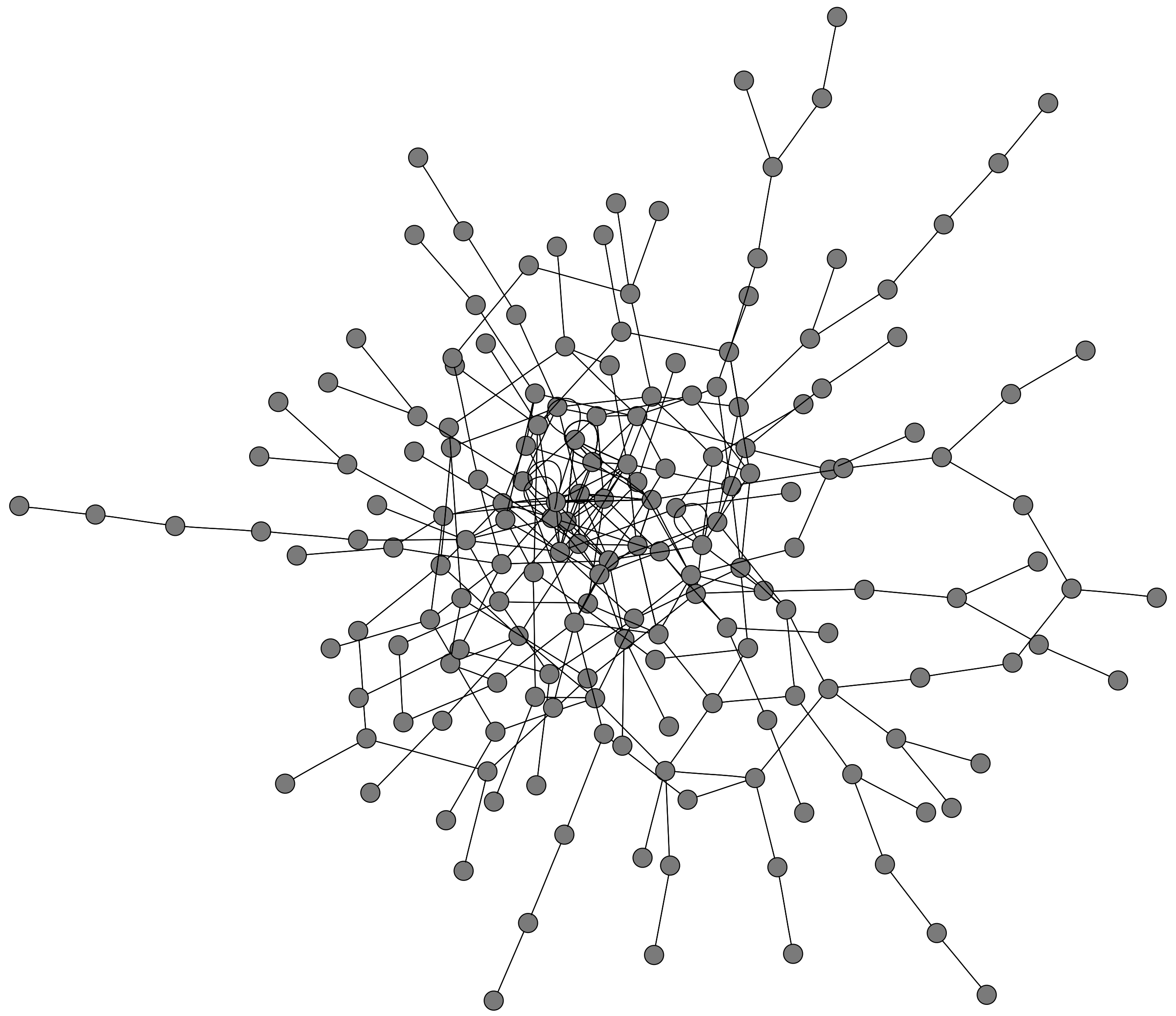
The assortativity coefficient is an example of a Pearson correlation coefficient, having a covariance in its numerator and a variance in the denominator.

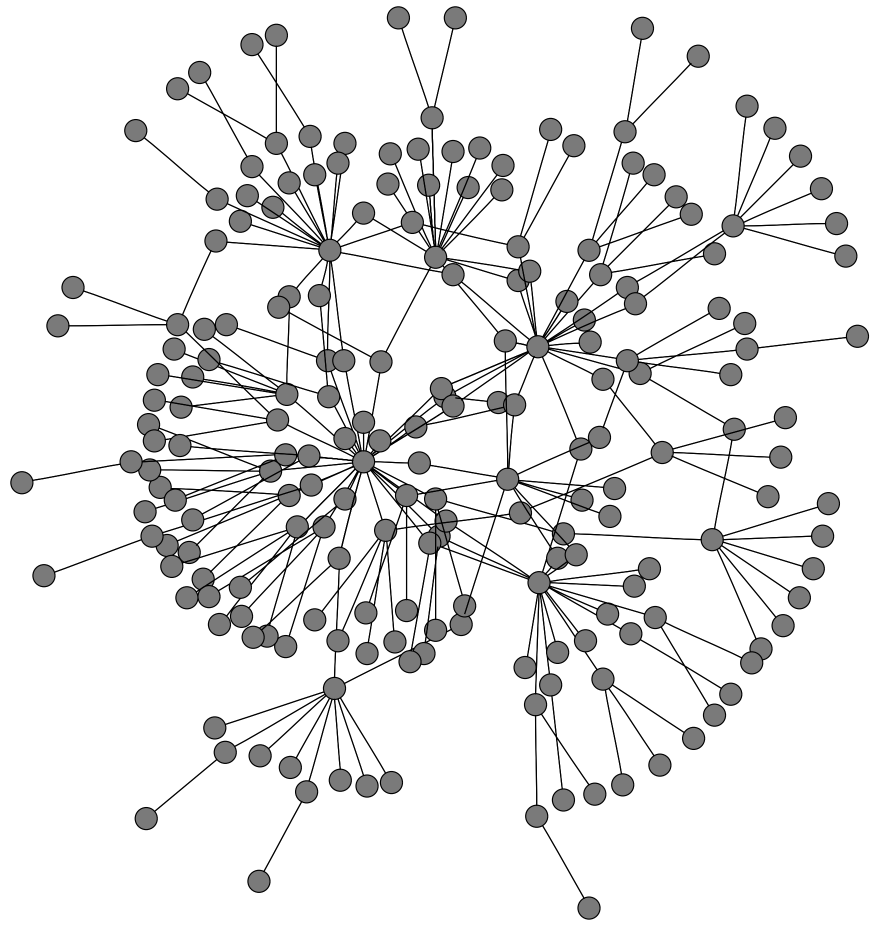
$$r = \frac{\sum_{ij}\left(A_{ij} - \frac{d_i\, d_j}{2m}\right) x_i\, x_j}{\sum_{ij}\left(d_i\delta_{ij} - \frac{d_i\, d_j}{2m}\right) x_i\, x_j}$$

The correlation coefficient varies between a maximum of 1 for a perfectly assortative network and a minimum of −1 for a perfectly disassortative one. E.g., the correlation coefficient of the example of the right takes a value of *r*=0.616, indicating that the friendship network has significant assortative mixing by age.



Homophily and Assortative Mixing • by Degree





A special case of assortative mixing according to a scalar quantity, and one of particular interest, is that of mixing by degree. In a network that shows assortative mixing by degree, the high-degree nodes will be preferentially connected to other high-degree nodes, and the low to low.

The reason this case is particularly interesting is because, unlike age or income, degree is itself a property of the network structure.

In particular, in an assortative network, where the high-degree nodes tend to stick together, one expects to get a clump or *core* of such high-degree nodes in the network surrounded by a less dense *periphery* of nodes with lower degree. This is represented by the network on the right, top-half.

Homophily and Assortative Mixing • by Degree

On the other hand, if a network is disassortatively mixed by degree, then high-degree nodes tend to be connected to low-degree ones, creating star-like features in the network that are often readily visible. This is represented by the network bottom-half of the figure on the right. Degree-disassortative networks do not usually have a core–periphery split but are instead more uniform.