Networks
Clustering, similarity and modularity

Elsa Arcaute

CASA0002: Urban Simulation

Week 7: 26th February 2024

Groups and Communities

→ Networks present sometimes a natural division into groups: friendship, similarity measures, chemical interchange among molecules, etc.

→ We would like to know whether these groups are the outcome of some particular process, or whether they could have formed at random

- → Does the particular interaction representing the link carry a special meaning for the network in question?
- → Could we get the same sort of grouping if a different measure was considered for the link, and hence a different network for the same population?

→ We might be interested in knowing how likely it is for groups to form in a network constructed in a specific way: **Global level**

→ Or we might be interested in comparing individuals within the system: **Local level**

Clustering coefficient

Transitivity

If A>B and B>C then A>C

What if '>' is any other sort of relationship?

e.g. If A is a friend of B and B is a friend of C, how likely it is for A to be a friend of C?

1) Find all the **possible** relations of transitivity in the system: **connected triples**
2) Count all the triangles in the system

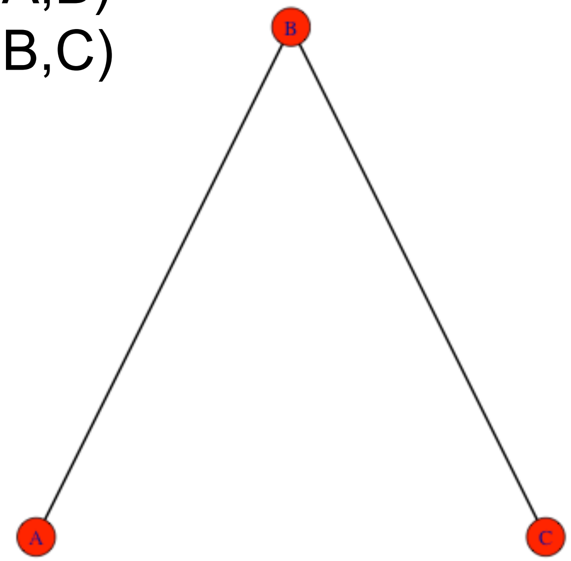
Connected triples:
Given 3 nodes A, B and C, 3 connected triplets can be formed

Clustering coefficient

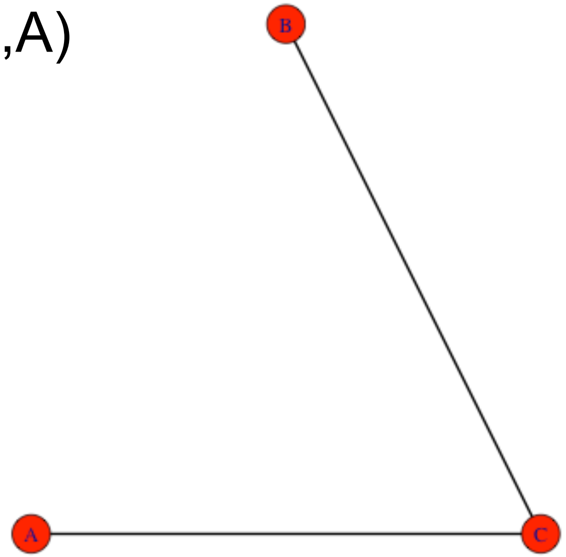
1) Find all the **possible** relations of transitivity in the system: **connected triples**
2) Count all the triangles in the system

Connected triples:
Given 3 nodes A, B and C, 3 connected triplets can be formed

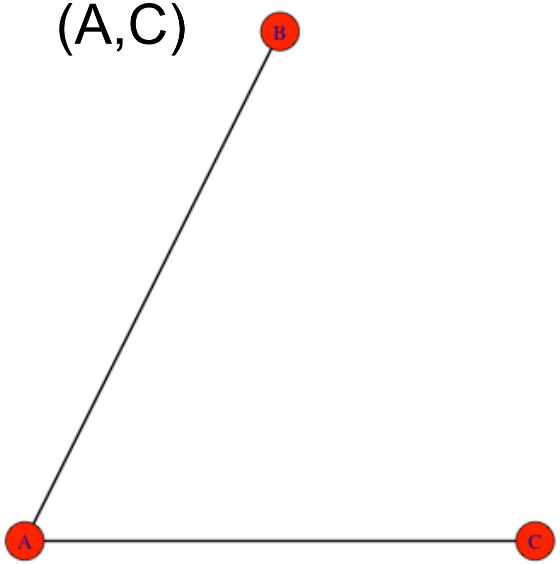
ABC
(A,B)
(B,C)



BCA
(B,C)
(C,A)



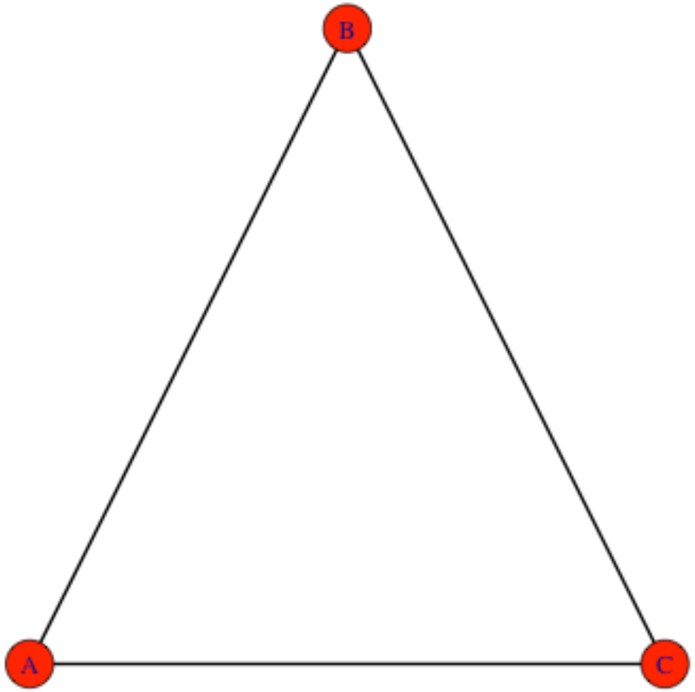
BAC
(B,A)
(A,C)



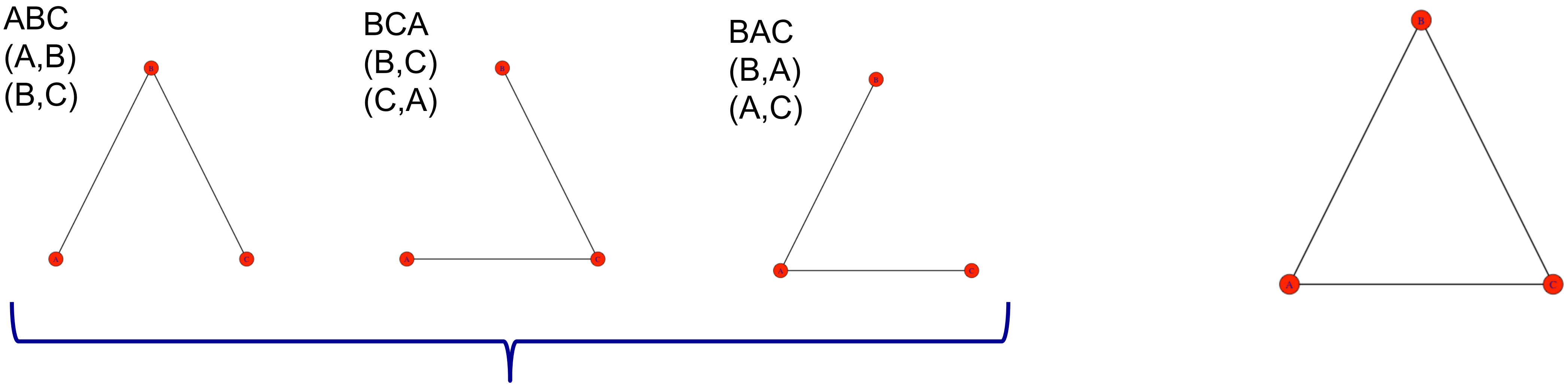
Clustering coefficient

1) Find all the **possible** relations of transitivity in the system: **connected triples**
2) Count all the triangles in the system

Triangles:
Given 3 nodes, a single triangle can be formed



Clustering coefficient



3 connected triples

1 triangle

$$C = \frac{(\text{Number of triangles}) \times 3}{(\text{number of connected triplets})}$$

Global

→ C in [0,1]

Clustering coefficient

$$C = \frac{\text{(Number of triangles) x 3}}{\text{(number of connected triplets)}}$$

Global

→ C in [0,1]

→ Is C big or small? Compare with random case!

E.g. Given a population of N individuals, let each individual have f friends. If A has f friends, among which B is his friend and picks f friends at random from the N population, the probability that one of the f friends of B is also a friend of A is: f/N.

For the network of film actor collaborations: C=0.20, while f/N=3.10^{-4}

→ So in this case C is very large!!! Much larger than expected from random connections

→ More likely people will be friends if they both have a common friend than if not.

Local clustering

→ Clustering coefficient for a single node:

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

For a single vertex *i*

→ **Average probability that a pair of *i*'s friends are friends of one another**

→ How does the degree of a node affect the clustering coefficient of the node?

→ What are the implications of a low local clustering coefficient for the passing of information? What about for the individual node?

Local clustering

- If we need the information to be passed as efficiently as possible, do we need high or low local clustering coefficients?

  HIGH

  - High clustering coefficient → resilience of spread of information

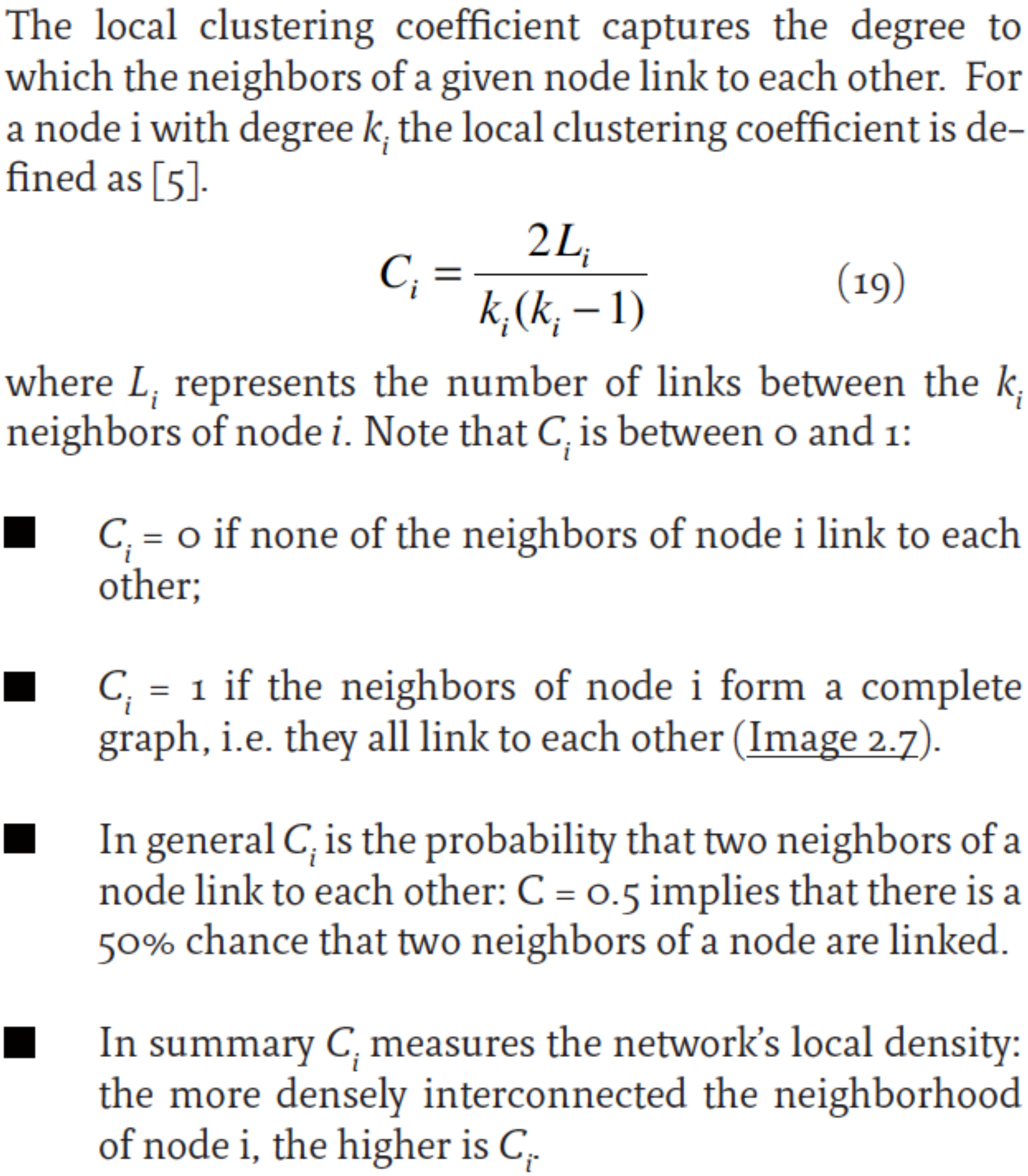
  - Low local clustering coefficients → *structural holes*=lack of connectivity between neighbours

- If I want to hold a position of power with respect to the control of information I need a low or a high clustering coefficient?

  LOW

  - Lack of connectivity between neighbours → broker between groups → recall betweenness centrality, this is a local version

Watts, Dodds, Newman, Science 296, 2002



The local clustering coefficient captures the degree to which the neighbors of a given node link to each other. For a node i with degree k_i the local clustering coefficient is defined as [5].

$$C_i = \frac{2L_i}{k_i(k_i - 1)} \qquad (19)$$

where L_i represents the number of links between the k_i neighbors of node *i*. Note that C_i is between 0 and 1:

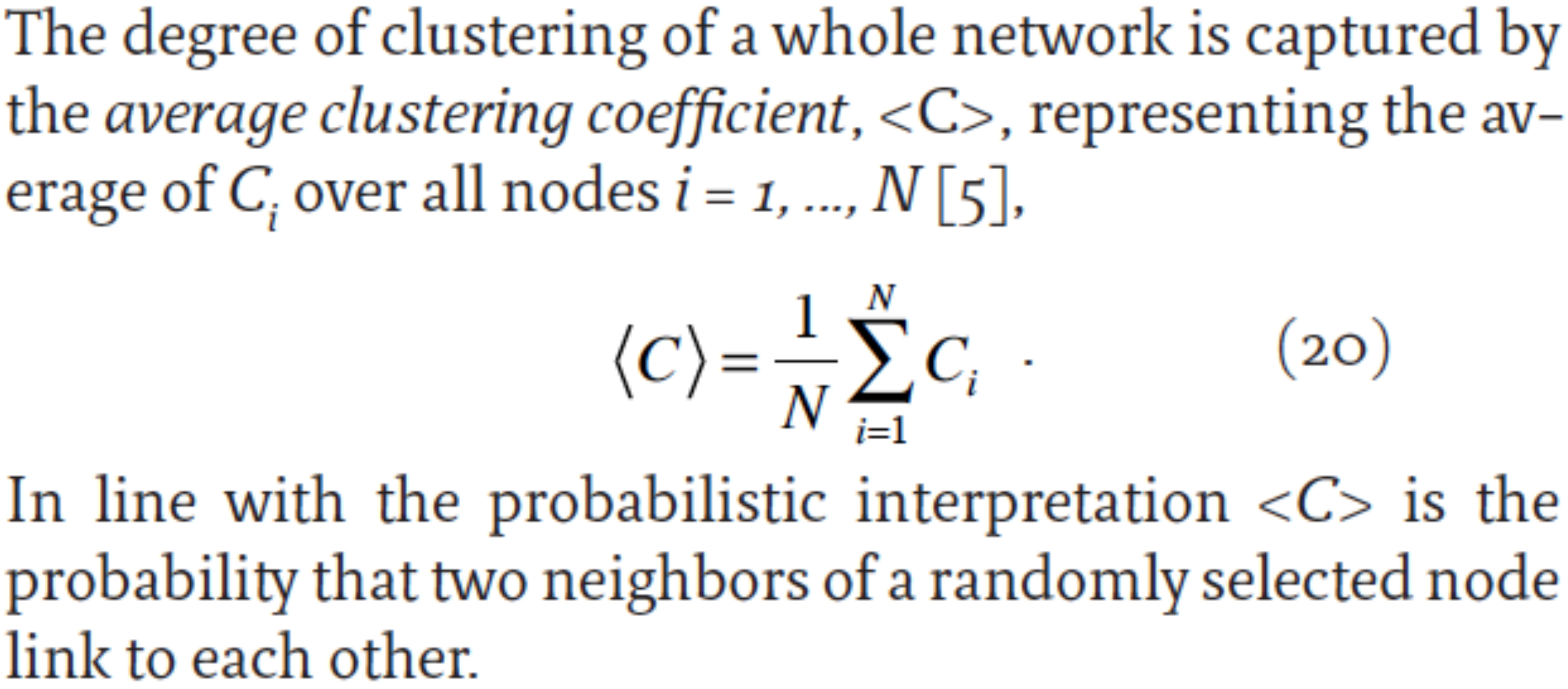
- C_i = 0 if none of the neighbors of node i link to each other;
- C_i = 1 if the neighbors of node i form a complete graph, i.e. they all link to each other (Image 2.7).
- In general C_i is the probability that two neighbors of a node link to each other: C = 0.5 implies that there is a 50% chance that two neighbors of a node are linked.
- In summary C_i measures the network's local density: the more densely interconnected the neighborhood of node i, the higher is C_i.

Slide from Barábasi's book

The total number of pairs of friends of *i* is:

$$Np=(1/2)\ k_i(k_i-1)$$

Slide from Barábasi’s book

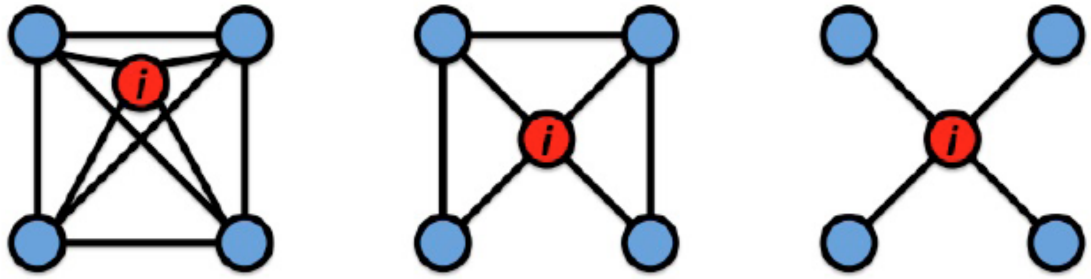


The degree of clustering of a whole network is captured by the *average clustering coefficient*, <C>, representing the average of C_i over all nodes $i = 1, \ldots, N$ [5],

$$\langle C \rangle = \frac{1}{N} \sum_{i=1}^{N} C_i \quad . \qquad (20)$$

In line with the probabilistic interpretation <C> is the probability that two neighbors of a randomly selected node link to each other.

Slide from Barábasi’s book

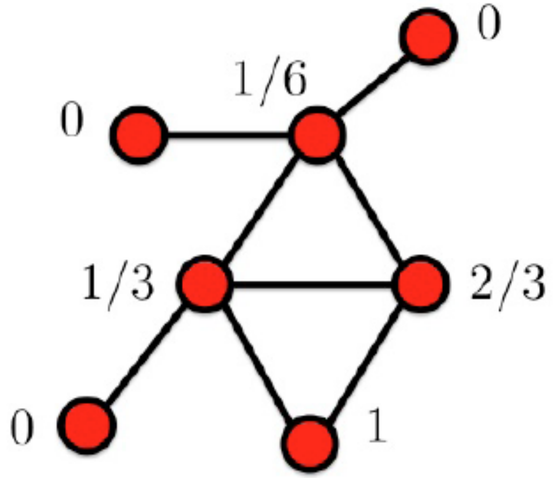


$C_i = 1$ $C = 1$

$C_i = 1/2$ $C = 9/14$

$C_i = 0$ $C = 0$

$$C = \frac{\text{(Number of triangles) x 3}}{\text{(number of connected triplets)}}$$



$$\langle C \rangle = \frac{13}{42} \approx 0.310$$

$$C = \frac{3}{8} = 0.375$$

$$C_i = \frac{\text{(n. connected pairs } i\text{'s nbrs)}}{\text{(n. pairs of neighbours of } i\text{)}}$$

Image 2.15

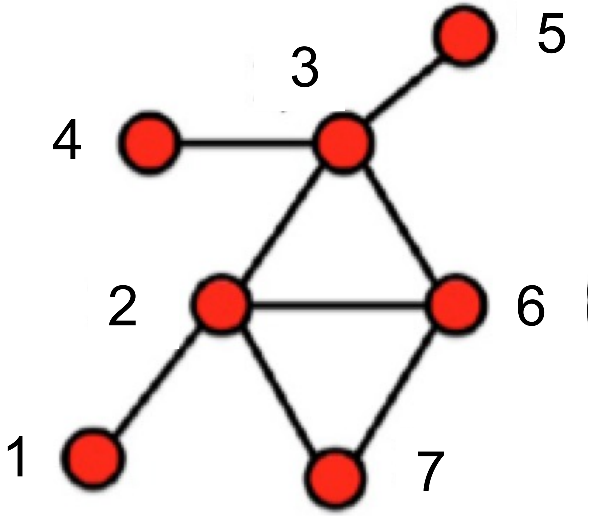
Clustering Coefficient.

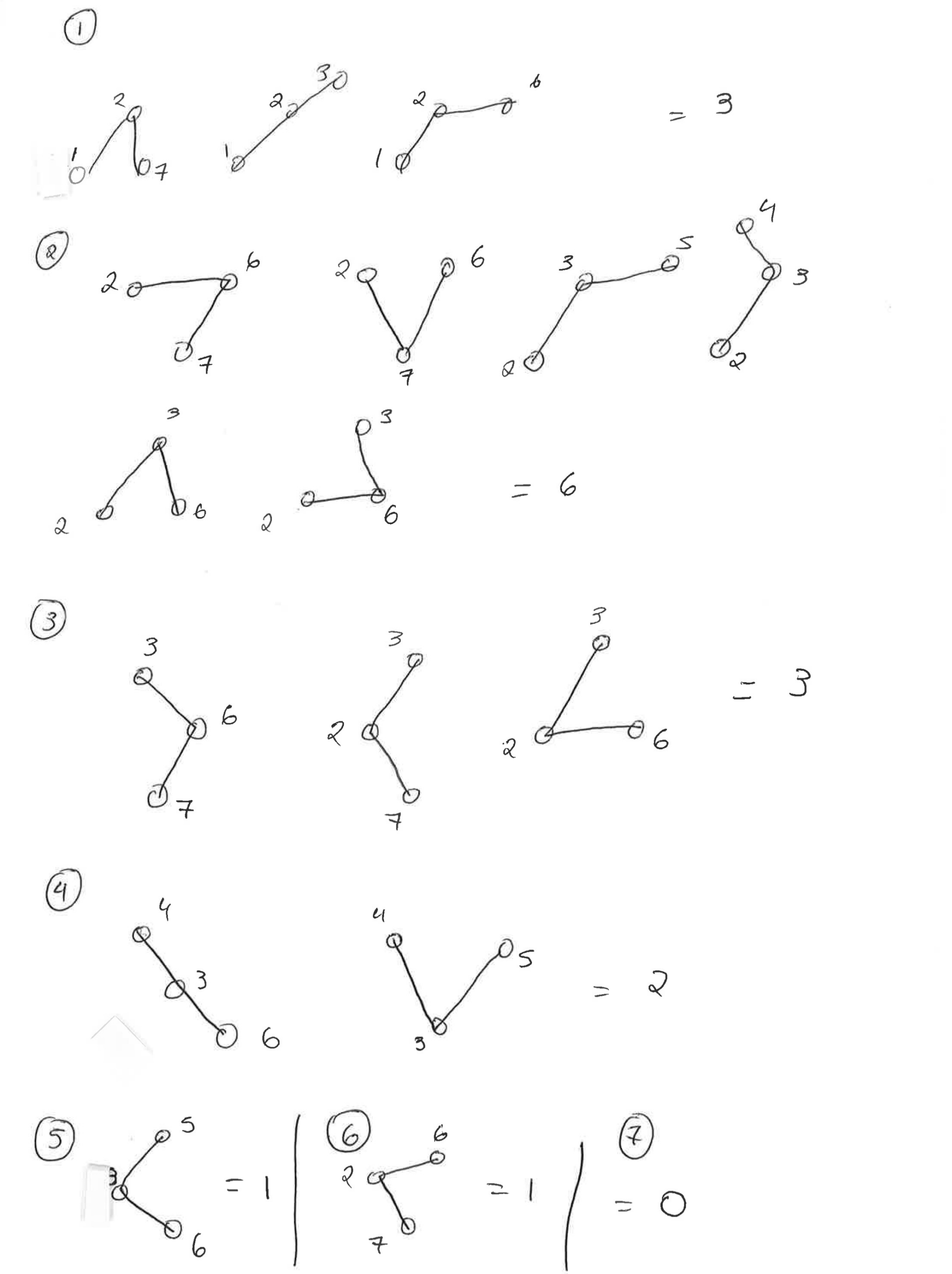
The local clustering coefficient, C_i, of the central node with degree k_i=4 for three different configurations of its neighborhood. The clustering coefficient measures the local density of links in a node's vicinity. The bottom figure shows a small network, with the local clustering coefficient of a node shown next to each node. Next to the figure we also list the network's average clustering coefficient <C>, according to Eq. (20), and its global clustering coefficient C, declined in Appendix A, Eq. (21). Note that for nodes with degrees k_i=0,1, the clustering coefficient is taken to be zero.

$$C_i = \frac{2L_i}{k_i(k_i-1)}$$

L_i: n. links btw k_i nbrs of node i

Counting connected triplets for previous graph (image 2.15 of Barabasi's book)





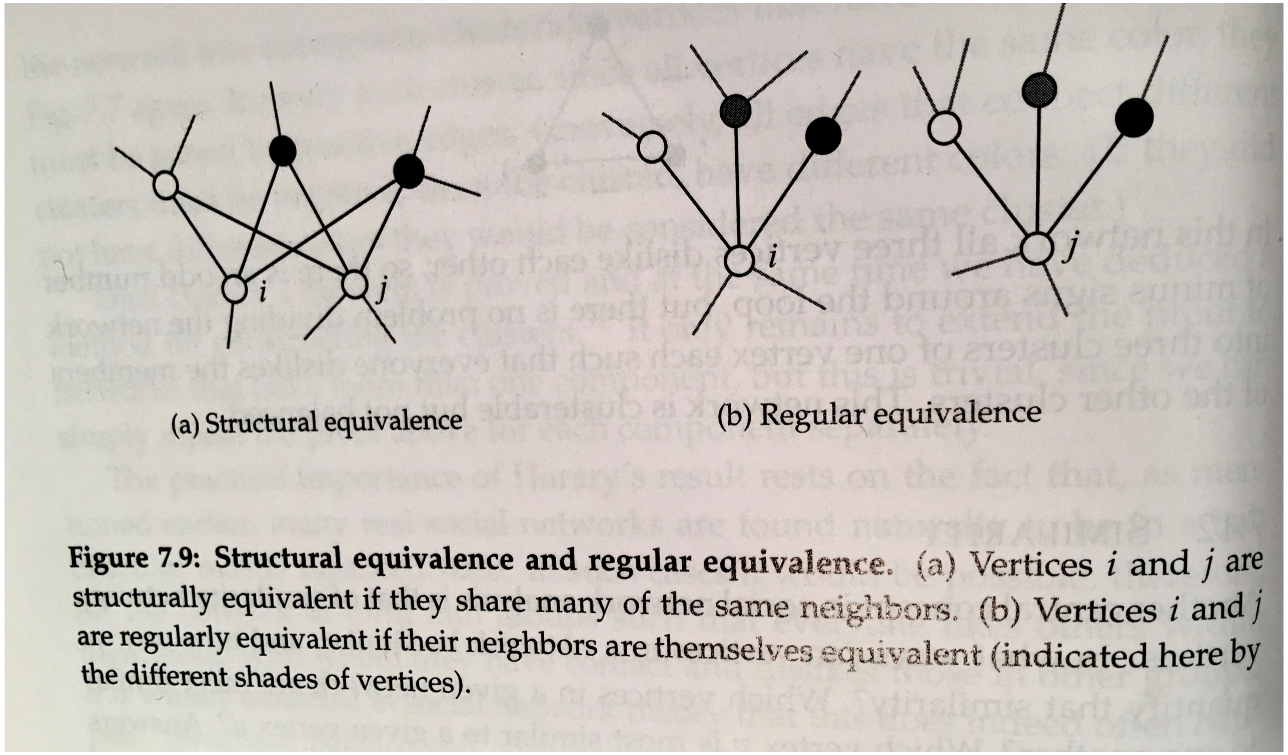
Similarity between nodes

→ If the nodes have similar friends, we might assume a measure of similarity between them.

Structural equivalence

→ If nodes play the same type of role in different networks, even though those nodes might seem unrelated between each other, they are similar with respect to their role in the system, say society, or a firm.

Regular equivalence



(a) Structural equivalence

(b) Regular equivalence

Figure 7.9: Structural equivalence and regular equivalence. (a) Vertices *i* and *j* are structurally equivalent if they share many of the same neighbors. (b) Vertices *i* and *j* are regularly equivalent if their neighbors are themselves equivalent (indicated here by the different shades of vertices).

Cosine similarity

This measure relates to the similarity with respect to the characteristics of two agents (recall these can be individuals, or places, etc). The characteristics are encoded in two vectors, say $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$: if the vectors overlap, the angle between them is 0, and *cos* $(\boldsymbol{\theta})=1$, while if they are completely different, they are perpendicular, and hence the angle is pi/2, and *cos* $(\boldsymbol{\theta})=0$:

$$\cos\theta = \frac{\vec{x_i} \cdot \vec{x_j}}{||\vec{x_i}||\,||\vec{x_j}||}$$

Note that in our case, we want to investigate whether nodes are similar with respect to whether they have a fair share of friends between them. In this case, the vector for node *i* corresponds to the ith row (or column) in the adjacency matrix, indicating the connections

$$\vec{x_i} = A_i$$

Cosine similarity

From the previous example, the norm of the vector is given by the square root of the degree of the node

$$||\vec{x_i}|| = \sqrt{k_i}$$

The inner product between the vectors is given by

$$\vec{x_i} \cdot \vec{x_j} = \sum_k A_{ik} A_{jk}$$

Hence the cosine similarity can be written as

$$\boxed{\sigma_{ij} = \frac{\sum_k A_{ik} A_{jk}}{\sqrt{k_i k_j}} = \frac{n_{ij}}{\sqrt{k_i k_j}}}$$

where n_{ij} are the common neighbours of *i* and *j*.

Pearson coefficients

Let us now compare the number of common friends compared to the possible ones if they were selected at random.

→ Node *i* chooses k_i neighbours at random from *n* possible individuals.

→ Node *j* also chooses k_j neighbours at random from *n* possible individuals.

→ The probability that the first person that *j* chooses is a friend of *i* is k_i/n.

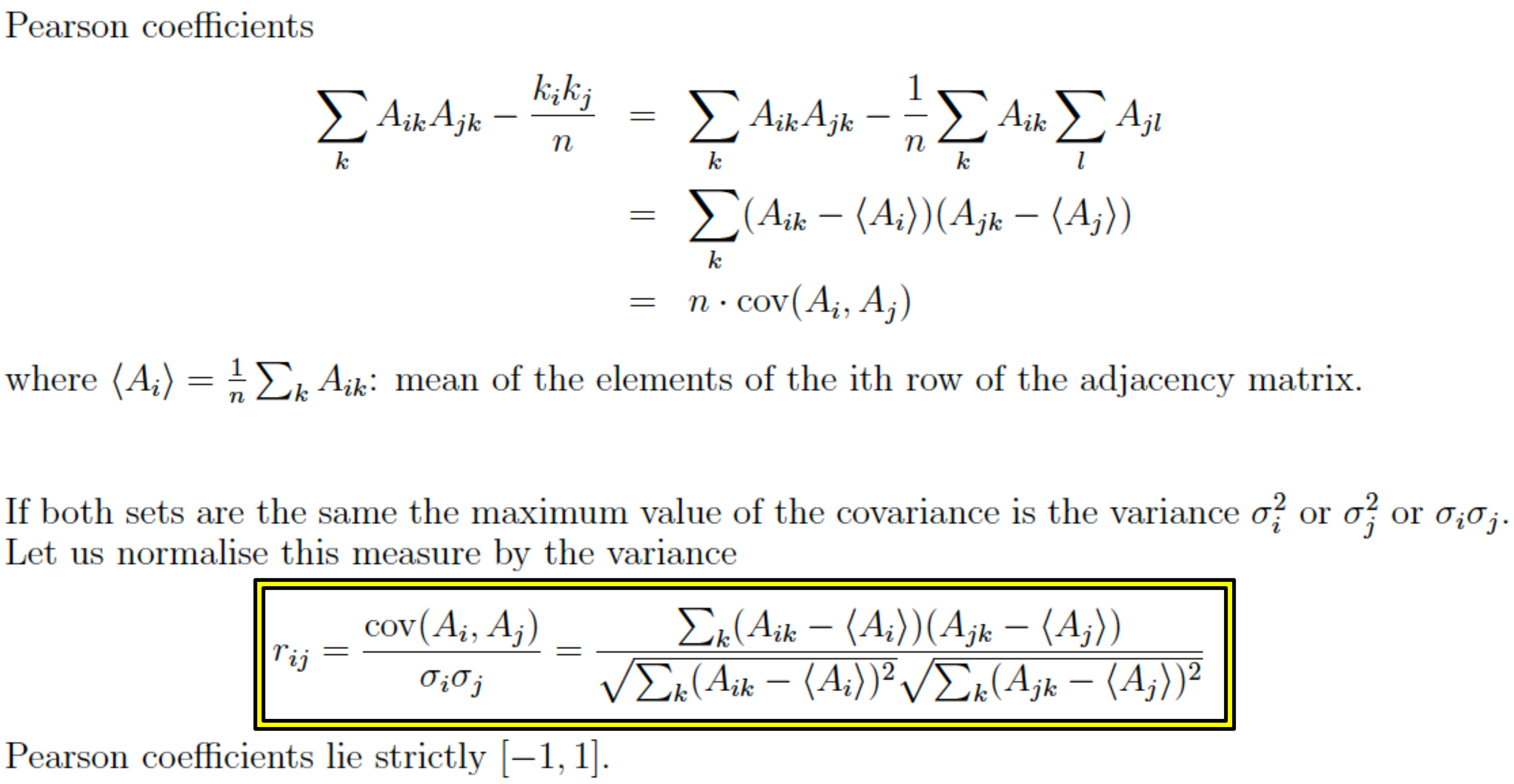
→ If chosen at random, the expected common friends between nodes *i* and *j* is $k_i k_j/n$.

→ Similarity between them can be

the number of friends – the expected number if chosen at random

$$\sum_k A_{ik} A_{jk} - \frac{k_i k_j}{n}$$

Pearson coefficients



Pearson coefficients

$$\begin{aligned}\sum_k A_{ik}A_{jk} - \frac{k_i k_j}{n} &= \sum_k A_{ik}A_{jk} - \frac{1}{n}\sum_k A_{ik}\sum_l A_{jl} \\ &= \sum_k (A_{ik} - \langle A_i\rangle)(A_{jk} - \langle A_j\rangle) \\ &= n \cdot \mathrm{cov}(A_i, A_j)\end{aligned}$$

where $\langle A_i\rangle = \frac{1}{n}\sum_k A_{ik}$: mean of the elements of the ith row of the adjacency matrix.

If both sets are the same the maximum value of the covariance is the variance σ_i^2 or σ_j^2 or $\sigma_i\sigma_j$. Let us normalise this measure by the variance

$$\boxed{r_{ij} = \frac{\mathrm{cov}(A_i, A_j)}{\sigma_i \sigma_j} = \frac{\sum_k (A_{ik} - \langle A_i\rangle)(A_{jk} - \langle A_j\rangle)}{\sqrt{\sum_k (A_{ik} - \langle A_i\rangle)^2}\sqrt{\sum_k (A_{jk} - \langle A_j\rangle)^2}}}$$

Pearson coefficients lie strictly $[-1, 1]$.

Constructing networks from properties of agents

Overall, for any type of system, one can create relationships between agents, according to any type of property of interest. This property can relate to the **similarity** between agents, or its **dissimilarity**. Hence, we can think in terms of **distance** as well. Things that are very close will be more likely to interact, and will be assumed to be more similar. Recall Tobler's first law of Geography "*everything is related to everything else, but near things are more related than distant things*".

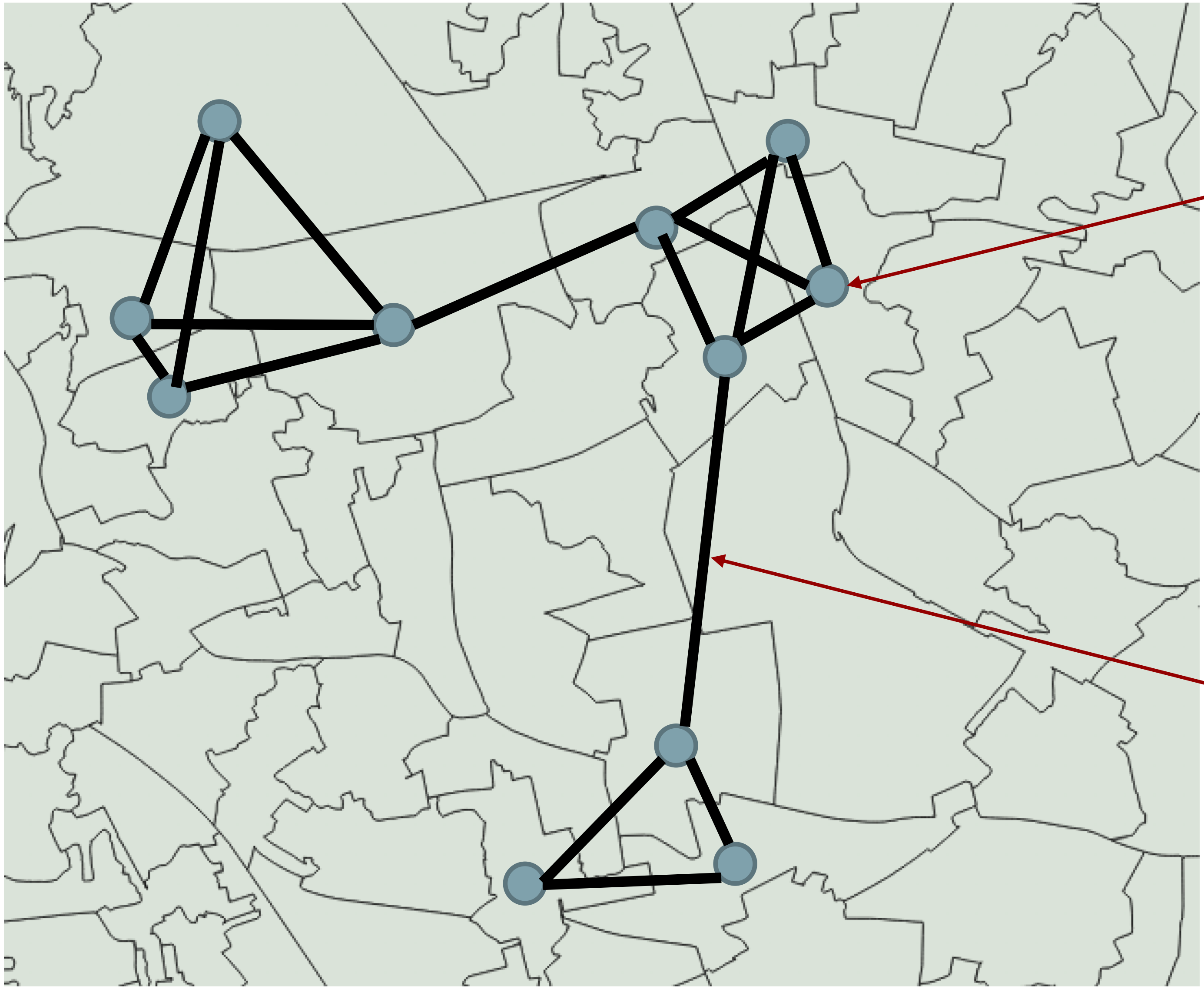
Take any system that you would like to analyse and create a network by assigning links according to a measure of distance between them

e.g.

→ street network can be converted into a network by considering the intersection points as nodes, and the streets as the links

→ For say census data, the centroids of the census tracks can act as nodes, and a measure of similarity can be computed by looking at the characteristics of the two areas and encoding them in vectors.

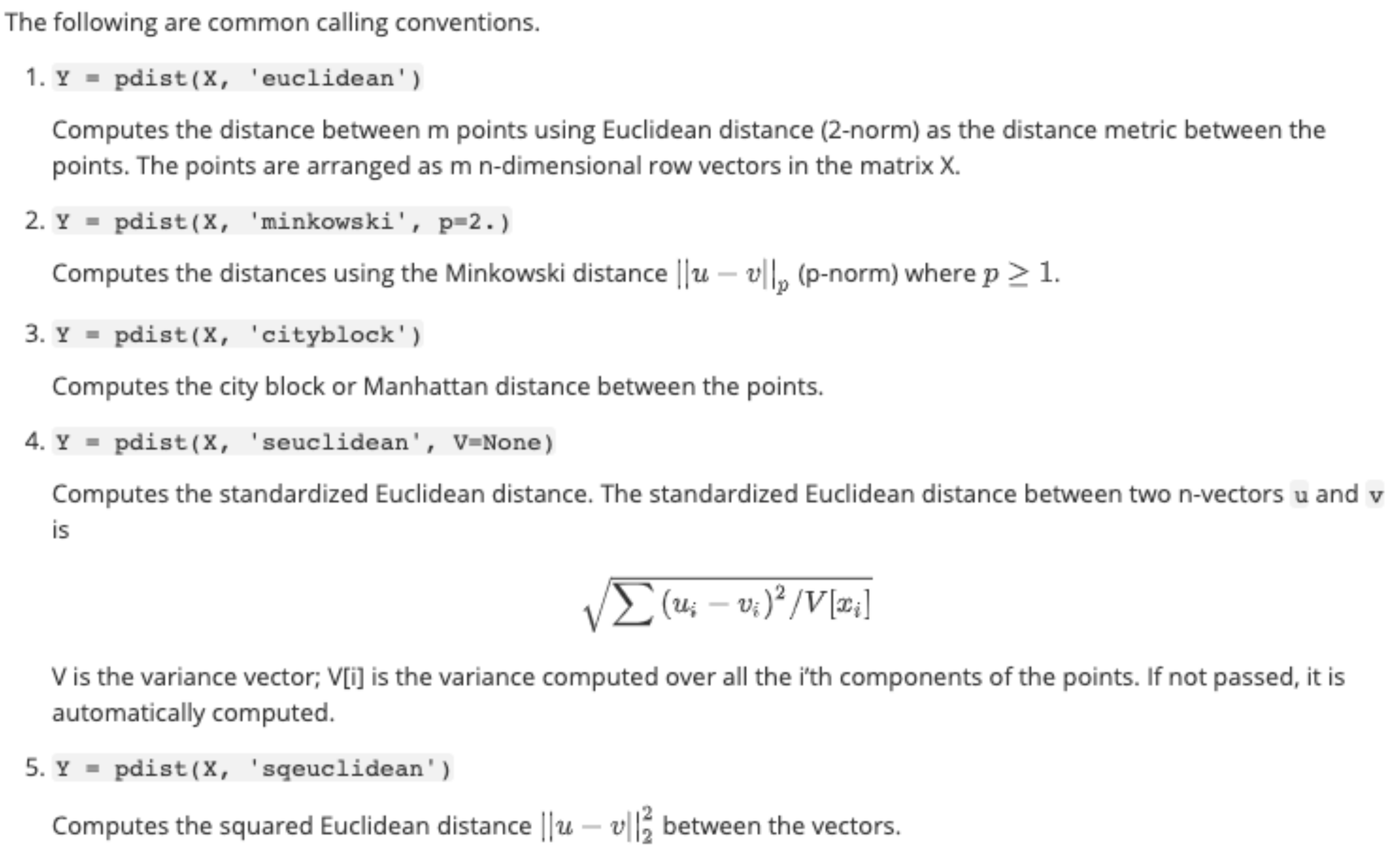
Network of relationships between places



Each area can be represented by its centroid, which will be considered as a node in the network

A link between the two areas is defined according to desired characteristic

Distance functions, examples in Python



The following are common calling conventions.

1. `Y = pdist(X, 'euclidean')`

   Computes the distance between m points using Euclidean distance (2-norm) as the distance metric between the points. The points are arranged as m n-dimensional row vectors in the matrix X.

2. `Y = pdist(X, 'minkowski', p=2.)`

   Computes the distances using the Minkowski distance $||u - v||_p$ (p-norm) where $p \geq 1$.

3. `Y = pdist(X, 'cityblock')`

   Computes the city block or Manhattan distance between the points.

4. `Y = pdist(X, 'seuclidean', V=None)`

   Computes the standardized Euclidean distance. The standardized Euclidean distance between two n-vectors `u` and `v` is

   $$\sqrt{\sum (u_i - v_i)^2 / V[x_i]}$$

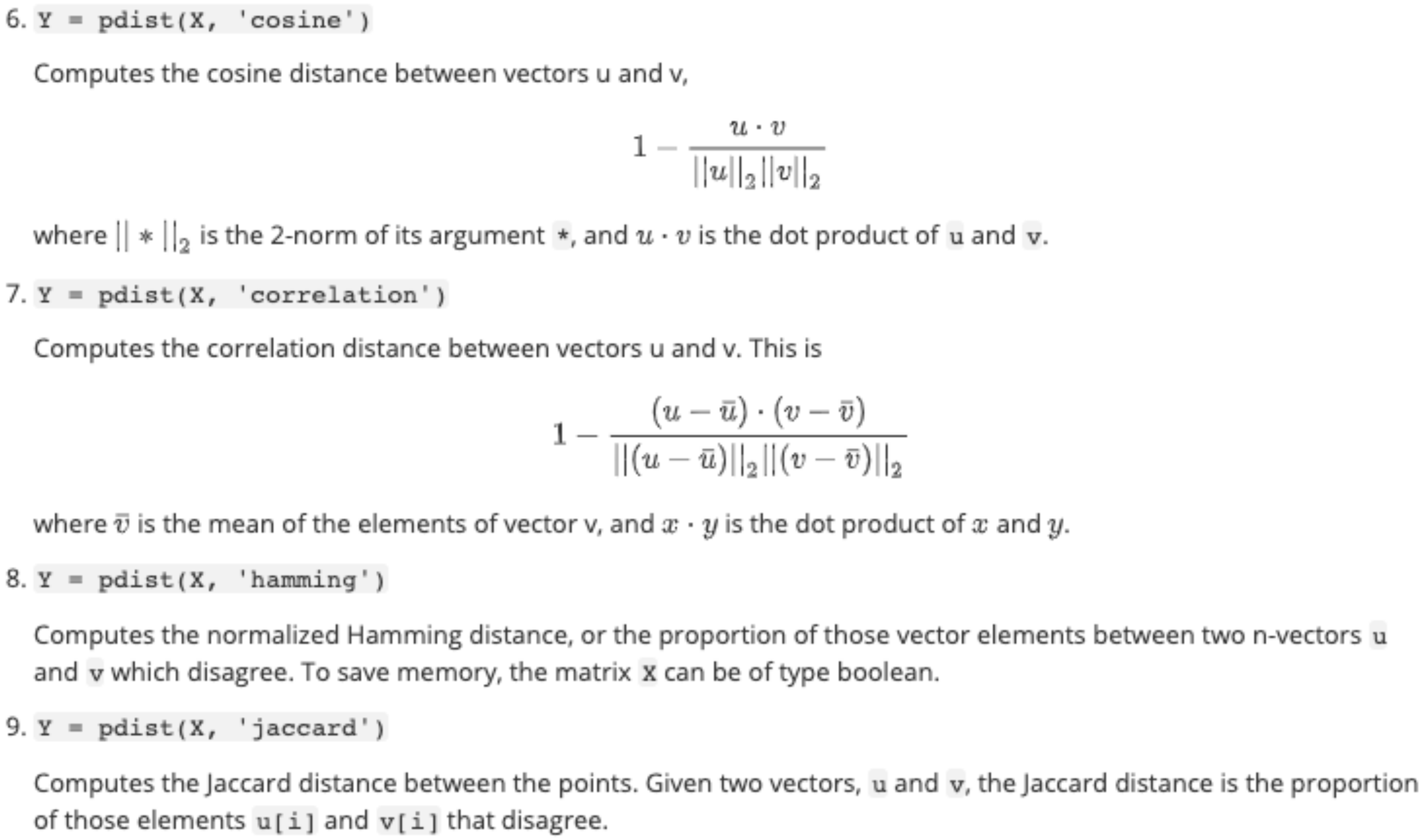
   V is the variance vector; V[i] is the variance computed over all the i'th components of the points. If not passed, it is automatically computed.

5. `Y = pdist(X, 'sqeuclidean')`

   Computes the squared Euclidean distance $||u - v||_2^2$ between the vectors.

https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.pdist.html

Distance functions, examples in Python



6. `Y = pdist(X, 'cosine')`

   Computes the cosine distance between vectors u and v,

   $$1 - \frac{u \cdot v}{||u||_2 ||v||_2}$$

   where $|| * ||_2$ is the 2-norm of its argument `*`, and $u \cdot v$ is the dot product of `u` and `v`.

7. `Y = pdist(X, 'correlation')`

   Computes the correlation distance between vectors u and v. This is

   $$1 - \frac{(u - \bar{u}) \cdot (v - \bar{v})}{||(u - \bar{u})||_2 ||(v - \bar{v})||_2}$$

   where $\bar{v}$ is the mean of the elements of vector v, and $x \cdot y$ is the dot product of x and y.

8. `Y = pdist(X, 'hamming')`

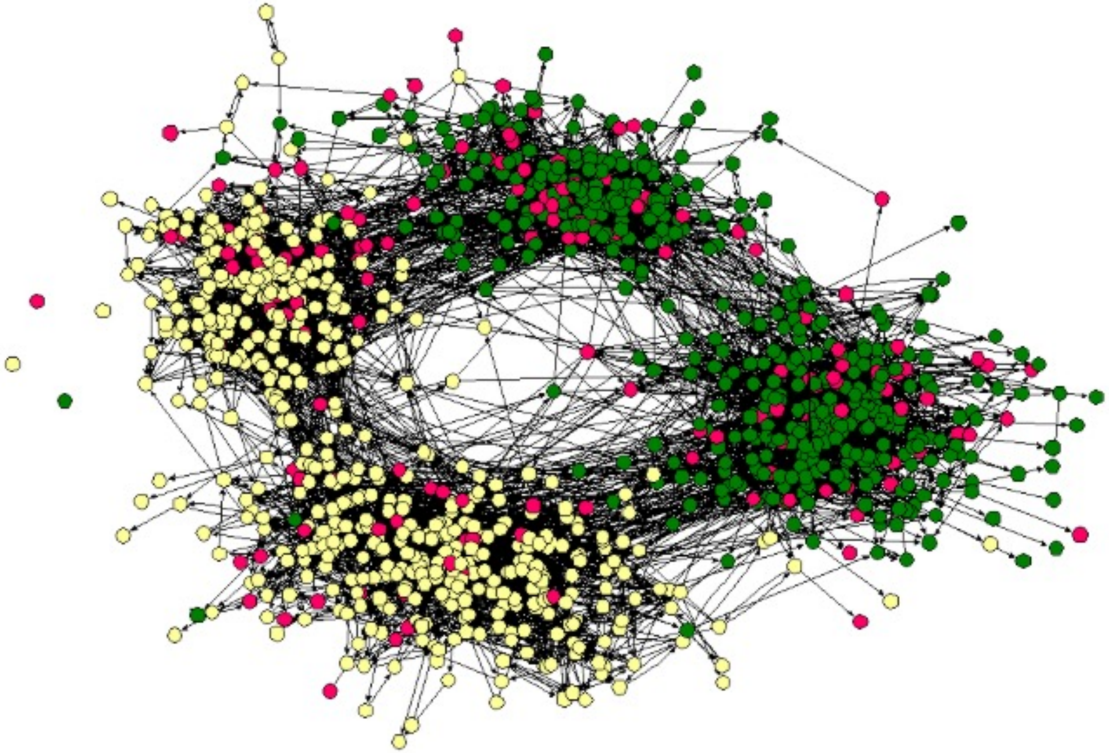
   Computes the normalized Hamming distance, or the proportion of those vector elements between two n-vectors `u` and `v` which disagree. To save memory, the matrix `X` can be of type boolean.

9. `Y = pdist(X, 'jaccard')`

   Computes the Jaccard distance between the points. Given two vectors, `u` and `v`, the Jaccard distance is the proportion of those elements `u[i]` and `v[i]` that disagree.

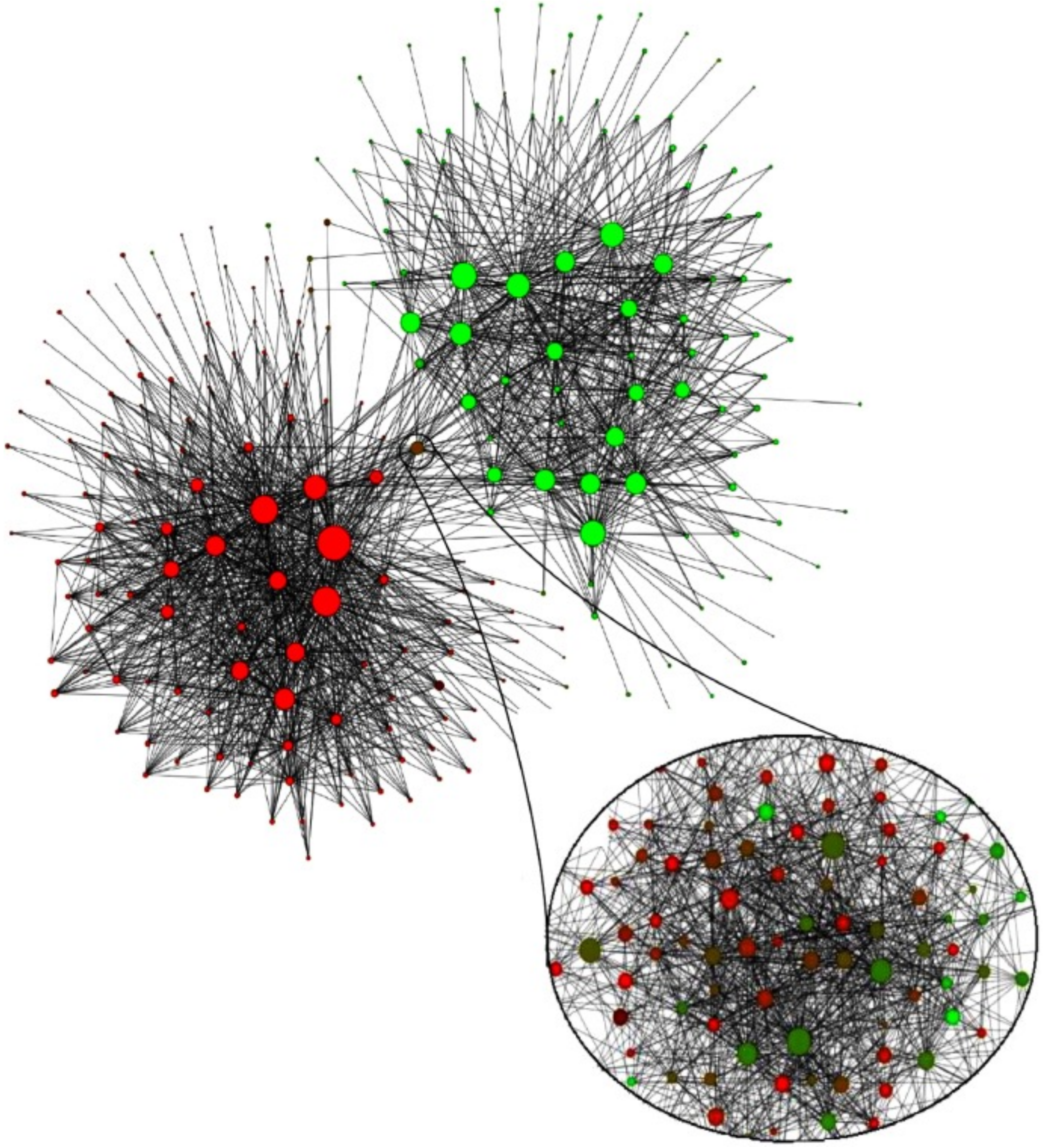
https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.pdist.html

Assortativity and modularity



Network of high school frienships

J. Moody, *Am. J. of Soc.* 107, 679-716, 2001.



Belgian mobile phone data

V. Blondel, J-L. Guillaume, R. Lambiotte and E. Lefebvre, *J. Stat. Mech.*, P10008, 2008.

Assortativity and modularity

People tend to group with people that share similarities

- High school network: different colour nodes represent different races
  → students were friends with students of a similar race

- Belgian mobile phone data: red nodes represent French speaking communities while green nodes represent Dutch speaking communities
  → Clear grouping pattern of grouping according to mainly spoken language

Homophily or assortative mixing

Clustering according to dissimilarities

→ e.g. Sexual contact networks: most likely a couple will be formed by opposite sex partners

Disassortative mixing

Assortativity and modularity

Assortative mixing

- ➢ Nodes of similar degree will cluster together
  → e.g. network of co-authorship in academia, film actors collaboration, etc.

Disassortative mixing

- ➢ Nodes of high degree will cluster with nodes of low degree
  → e.g. Neural networks, protein interactions, the internet, etc

M.E.J. Newman, *Phys. Rev. Lett* 89, 208701 (2002)

Assortativity and modularity

How assortative is the network?

Need to **compare** the fraction of links of nodes of the same type with the fraction of links of nodes of the same type that would emerge from a **random structure**.

→ You always need to compare against the null model!!!!! In this case we do NOT expect that random networks will cluster in a specific way.

Assortativity can be given in terms of differentiable classes, or in terms of similarity:

- Classes: for non-scalar characteristics denoted by a type c_i for a node *i*; say gender, or language, etc.,

- Similarity: for scalar values x_i of a specific characteristic; say socio-economic attribute, such as income, demographic attribute, such as age, etc.

M.E.J. Newman *PNAS* 103 (2006)

Assortativity and modularity

How assortative is the network?

- Classes: for non-scalar characteristics denoted by a type c_i for a node *i*; say gender, or language, etc., the total n. of links for nodes of same type is

$$\sum_{\text{links }(i,j)} \delta(c_i, c_j) = \frac{1}{2} \sum_{ij} A_{ij} \delta(c_i, c_j)$$

where

$$\delta(c_i, c_j) = \begin{cases} 1 & \text{if } c_i = c_j, \\ 0 & \text{otherwise} \end{cases}$$

In a random network, a node of degree k_i has a probability of $k_j / 2m$ to be attached to a node of degree k_j , where *m* is the total n. of links → $k_i k_j / 2m$ total n. of links between *i* and *j*.

M.E.J. Newman *PNAS* 103 (2006)

Assortativity and modularity

Modularity: discrete case

In order to understand if the observed network is special, or the grouping observed would have happened anyway given the number of nodes and links, we need to compare it with the random case. Recall that in a random network, a node of degree k_i has a probability of $k_j / 2m$ to be attached to a node of degree k_j , where m is the total n. of links → $k_i k_j /2m$ total n. of links between i and j.

Prob. of links in network

Expected prob of links
IF network is random

- Modularity matrix

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

M.E.J. Newman *PNAS* 103 (2006)

Assortativity and modularity

Modularity: discrete case

Prob. of links in network

Expected prob of links
IF network is random

- Modularity matrix

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

- Modularity

$$Q = \frac{1}{2m} \sum_{ij} B_{ij} \delta(c_i, c_j)$$

If Q>0 assortative mixing
If Q<0 disassortative mixing

- Max modularity

$$Q_{max} = \frac{1}{2m} \left(2m - \sum_{ij} \frac{k_i k_j}{2m} \delta(c_i, c_j) \right)$$

Recall:

$$m = \frac{1}{2} \sum_{i=1}^{n} k_i = \frac{1}{2} \sum_{ij} A_{ij}$$

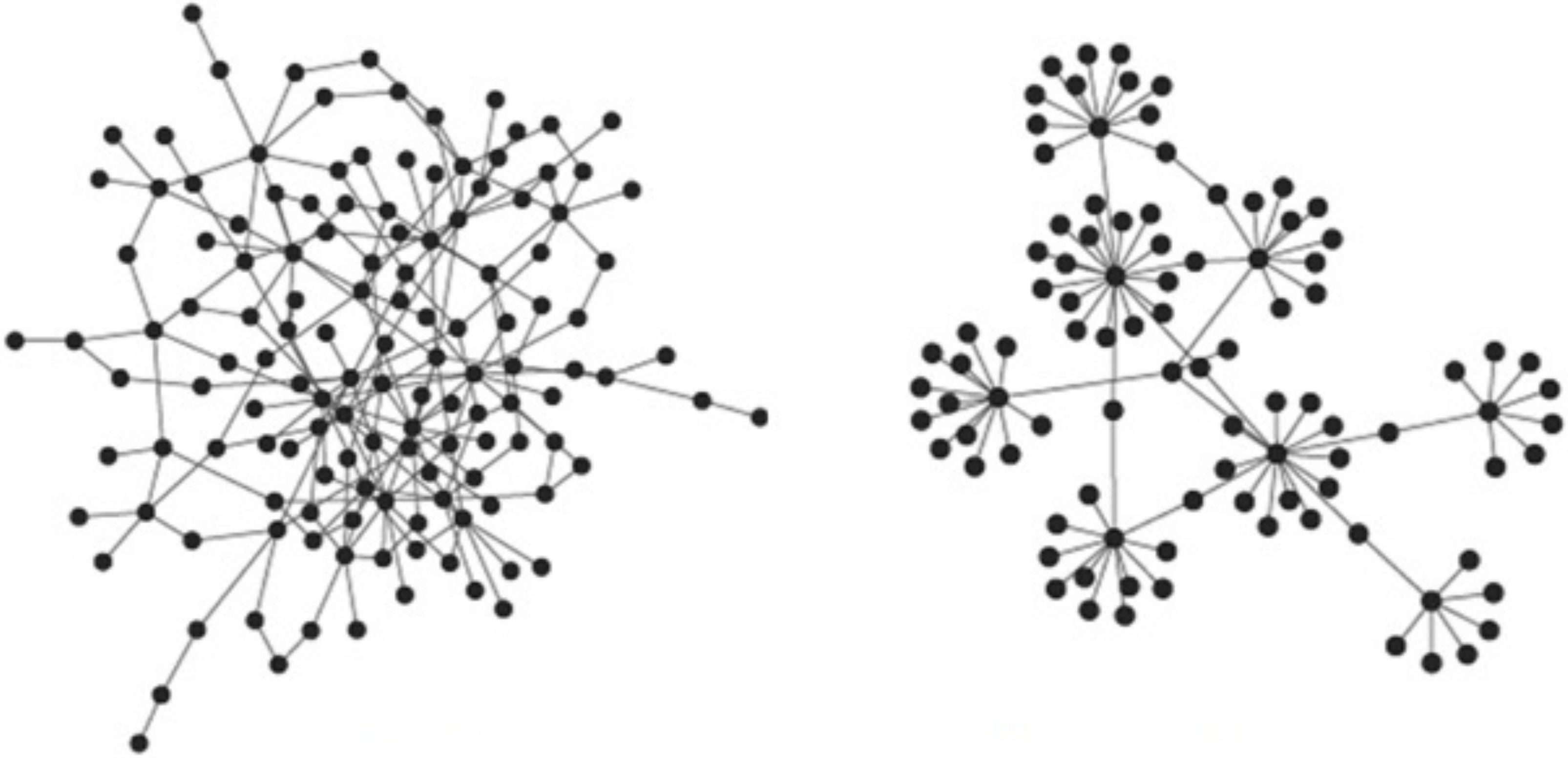
- Normalised modularity

or

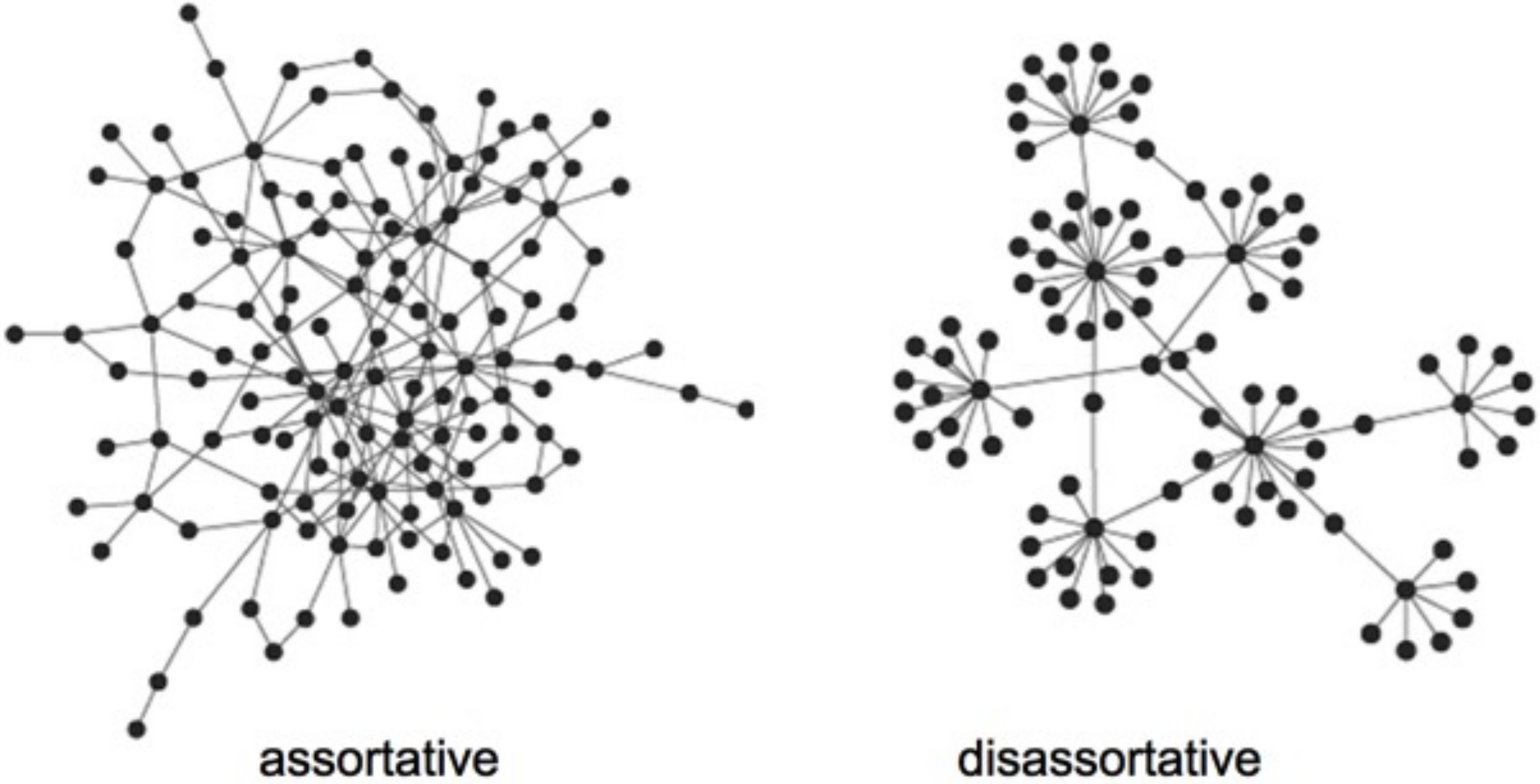
assortativity coefficient

$$M = \frac{Q}{Q_{max}}$$

M.E.J. Newman *PNAS* 103 (2006)



Newman and Girvan, Stat. Mech. of Complex Networks, 2003



Newman and Girvan, Stat. Mech. of Complex Networks, 2003

Assortativity and modularity

Modularity: similarity case

- Replace $\delta(c_i, c_j)$ with values $x_i x_j$

→ get covariance

$$\mathrm{cov}(x_i, x_j) = \frac{1}{2m} \sum_{ij} B_{ij} x_i x_j$$

- Max cov for $xi=xj$

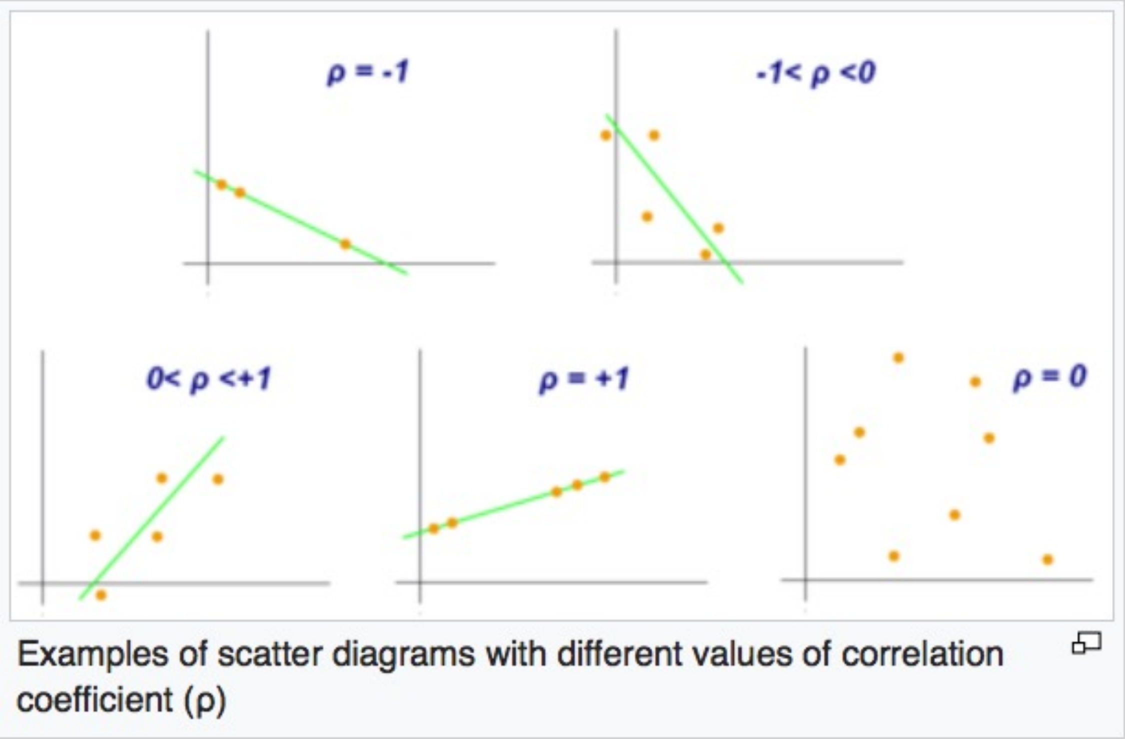
→ ***assortativity coefficient***

$$\boxed{r = \frac{\sum_{ij} B_{ij} x_i x_j}{\sum_{ij} (k_i \delta_{ij} - k_i k_j / 2m) x_i x_j}}$$

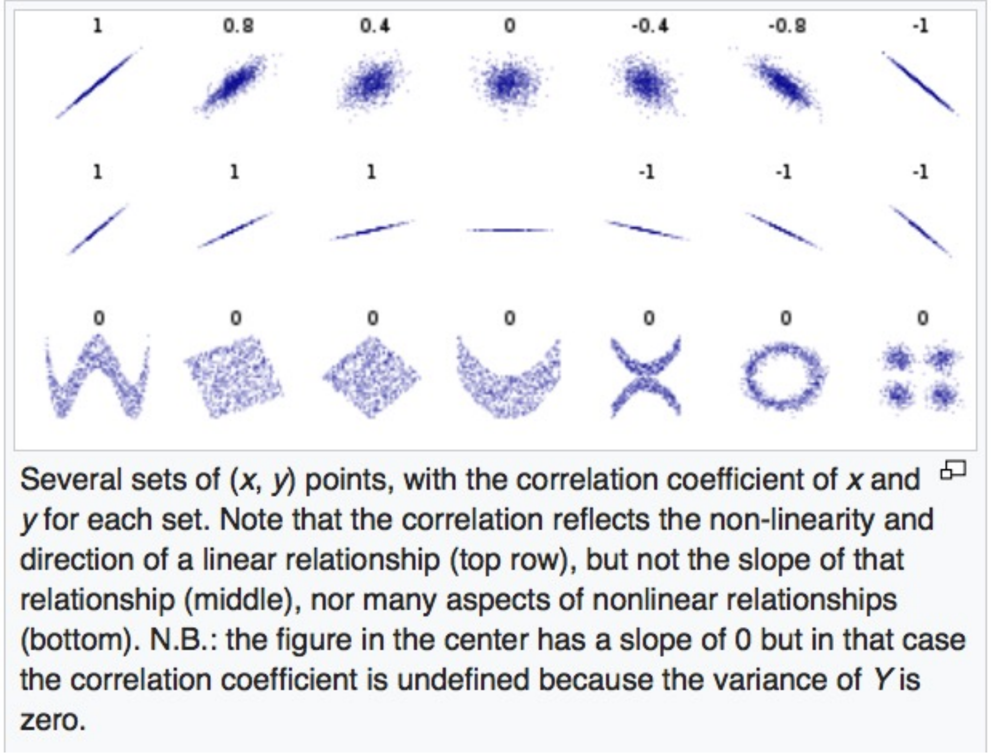
$$r = \begin{cases} 1 & \text{perfectly assortative,} \\ -1 & \text{perfectly disassortative,} \\ 0 & \text{random: no correlation between nodes} \end{cases}$$

M.E.J. Newman *PNAS* 103 (2006)

Recall Pearson correlation



Examples of scatter diagrams with different values of correlation coefficient (ρ)



Several sets of (*x*, *y*) points, with the correlation coefficient of *x* and *y* for each set. Note that the correlation reflects the non-linearity and direction of a linear relationship (top row), but not the slope of that relationship (middle), nor many aspects of nonlinear relationships (bottom). N.B.: the figure in the center has a slope of 0 but in that case the correlation coefficient is undefined because the variance of *Y* is zero.

Source: Wikipedia

Community detection algorithms → coming soon!

There are many different algorithms to separate nodes within a network into different communities. These are based on the following assumptions:

→ Modularity plays a fundamental role in understanding whether a network can be partitioned
  → the partitioning can be done such that the modularity is maximised

→ Communities can be understood in many different senses:
  → As a connected subgraph
  → As a dense neighbourhood of a network: e.g. random walker might spend a lot of time in a particular set of nodes

→ No community structure is expected from a random network

Coming soon: we will look at the properties of different community detection algorithms that can be found in the literature.