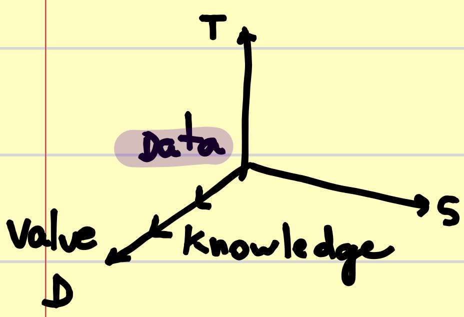
20241010_Business_Intelligence_WIN6

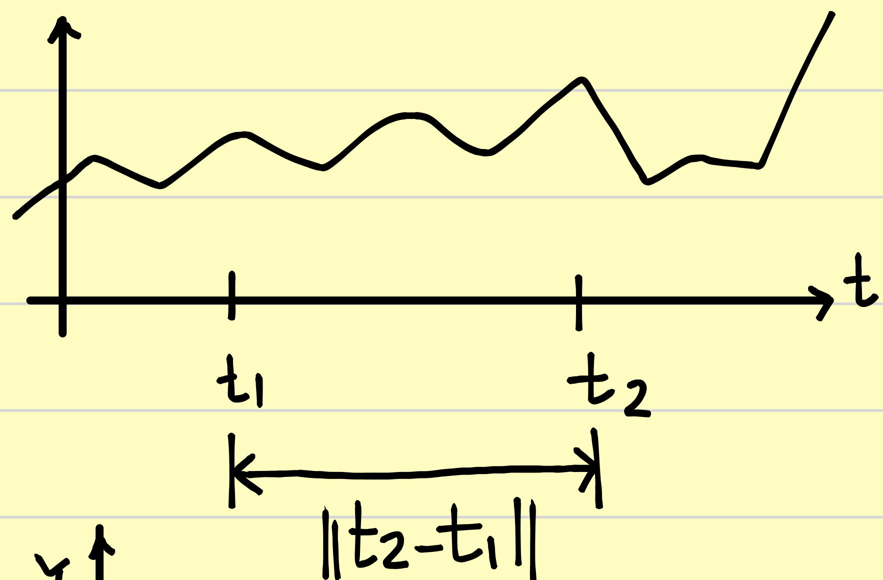


Datasets can be classified in 2 groups:

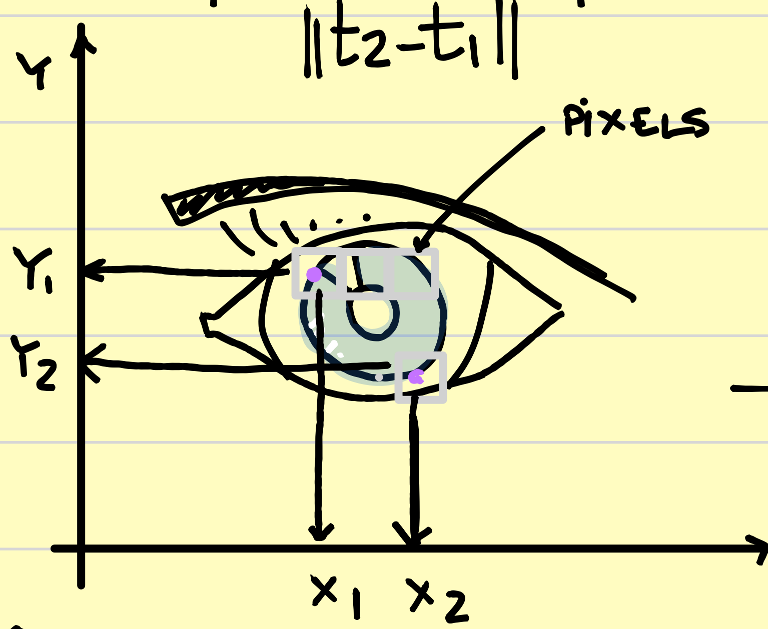
- EUCLIDEAN. if a metric can be defined to describe the distance.
- NON-EUCLIDEAN. otherwise.

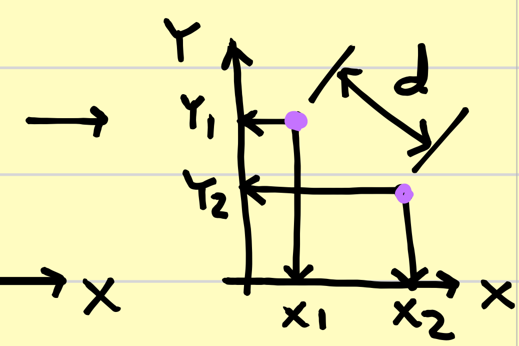
Examples of EUCLIDEAN Datasets:

- Timeseries (1D)



- Picture B&W (2D)





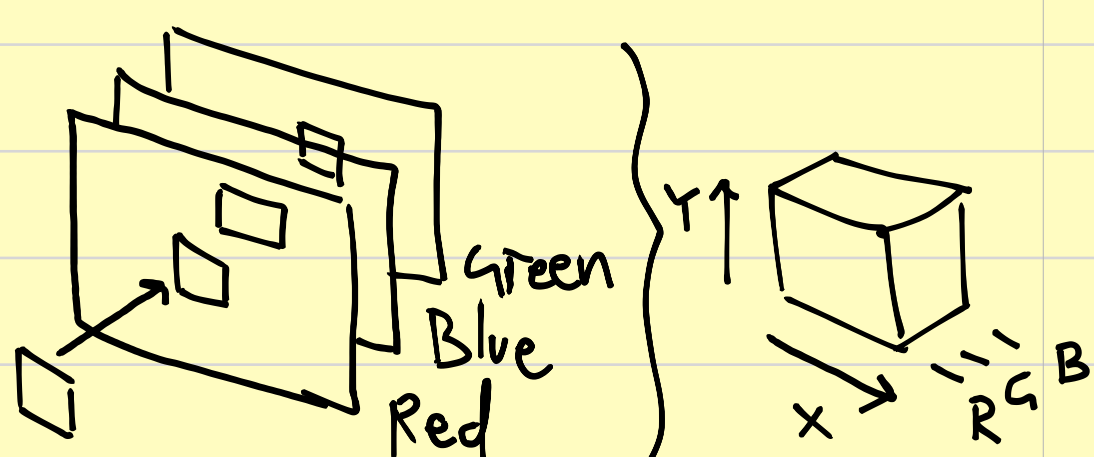
□ : [0,255]

8bits: 0 1 0 1 1 1 0 0

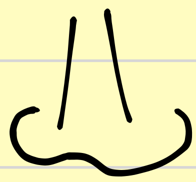
$2^8 = 256$

$$d = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

- Picture Color (3D)



R : [0,255]
B : [0,255]
G : [0,255]

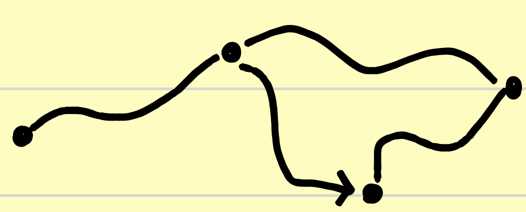
- Video Color
  (3D) + time + Sound → (5D)
-  Olfative
  (12D)
- Haptic
  (6D)

...

The hypothesis is that the underlying structure of the dataset allows for the calculation of a distance. This implies there is a way to "measure" between points in the space.

Example of non-euclidean datasets: NETWORKS.

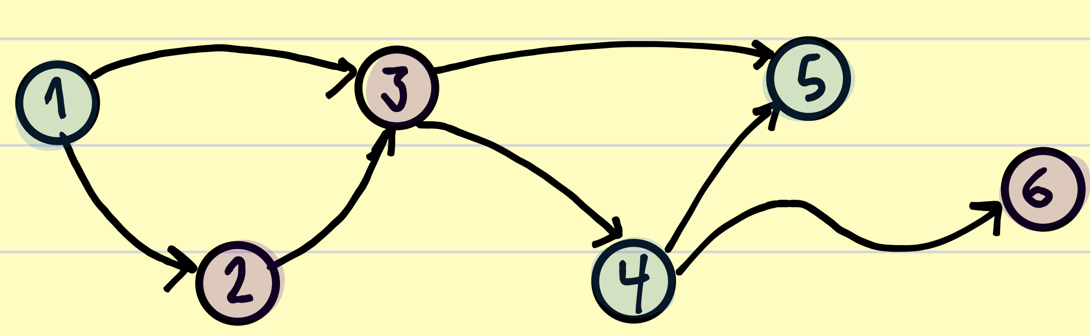
What is a network? Nodes + Edges + Attributes



A network is defined (mathematically) by a set (group) of nodes and a set of edges. This group of sets is called "GRAPH" (G).

Network ≡ GRAPH ≡ $G(N \cup E)$ [N. Nodes, E. Edges]

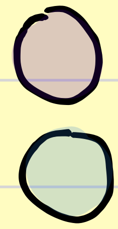
Example:



N: $\{1,2,3,4,5,6\}$ E: $\{(1\to2);(1\to3);(2\to3);$
$(3\to4);(3\to5);(4\to5);$
$(4\to6)\}$

$G=\{N \cup E\}$

Attributes:



Networks can be directed or undirected:

- Directed if the edges "have an arrow"
- Undirected otherwise

HOW CAN WE QUANTIFY & COMPARE NETWORKS?

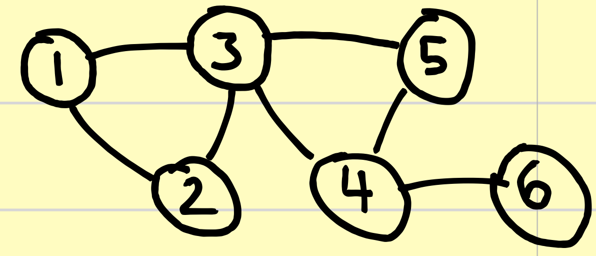
1. AVERAGE PATH LENGTH (APL)
2. CLUSTERING COEFFICIENT (CC)
3. LAPLACIAN of NETWORK

1. APL. Average distance (steps) between the nodes in the network.

$$APL = \frac{1}{N(N-1)} \cdot \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}$$

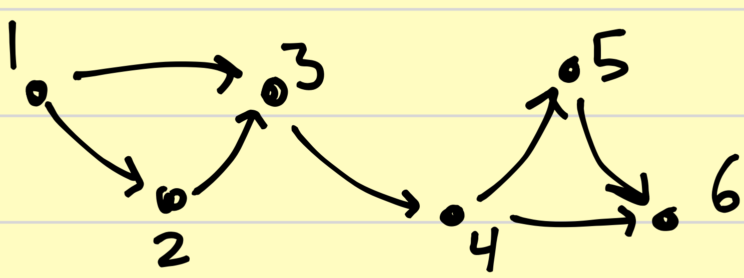
- The maximum # of relationships in a network of N elements is $N\cdot(N-1)$ because each element relates to all others, hence $(N-1)$, N times.
- $\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}$ represents the sum of all paths between nodes.

$$\begin{aligned} APL = \frac{1}{6.5}\cdot \Big[& {}^{1}\big[\overset{d_{12}}{1}+\overset{d_{13}}{1}+\overset{d_{14}}{2}+\overset{d_{15}}{2}+\overset{d_{16}}{3}\big] + \\ & {}^{2}\big[\overset{d_{21}}{1}+\overset{d_{23}}{1}+\overset{d_{24}}{2}+\overset{d_{25}}{2}+\overset{d_{26}}{3}\big] + \\ & {}^{3}\big[\overset{d_{31}}{1}+\overset{d_{32}}{1}+\overset{d_{34}}{1}+\overset{d_{35}}{1}+\overset{d_{36}}{2}\big] + \\ & {}^{4}\big[\overset{d_{41}}{2}+\overset{d_{42}}{2}+\overset{d_{43}}{1}+\overset{d_{45}}{1}+\overset{d_{46}}{1}\big] + \\ & {}^{5}\big[\overset{d_{51}}{2}+\overset{d_{52}}{2}+\overset{d_{53}}{1}+\overset{d_{54}}{1}+\overset{d_{56}}{2}\big] + \\ & {}^{6}\big[\overset{d_{61}}{3}+\overset{d_{62}}{3}+\overset{d_{63}}{2}+\overset{d_{64}}{1}+\overset{d_{65}}{2}\big]\Big] = \dots \end{aligned}$$



Example. Compare following two graphs in terms of performance related to APL.

G_1



APL_1

G_2

APL_2

> When you compare two networks, the one with the shortest APL would be usually faster/more effective.

2. CC Describes how good groups are created in the network.

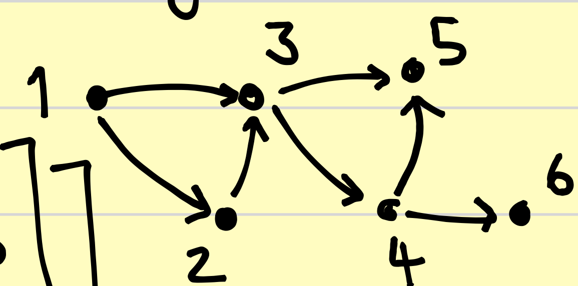
$$CC = \frac{1}{N} \sum_{i=1}^{N} \frac{2L_i}{k_i(k_i-1)}$$

$L_i \equiv$ Number of relationships between the neighbors of node "i"
(how many of my friends are friends to each other?)

$k_i \equiv$ Number of neighbors of node "i"

$$CC = \frac{1}{6}\left[{}^{1}\left[\frac{2 \cdot 1}{2(2-1)}\right] + {}^{2}\left[\frac{2 \cdot 1}{2(2-1)}\right] + {}^{3}\left[\frac{2 \cdot 2}{4(4-1)}\right] + {}^{4}\left[\frac{2 \cdot 1}{3(3-1)}\right] + {}^{5}\left[\frac{2 \cdot 1}{2 \cdot (2-1)}\right] + {}^{6}\left[\phi\right] \right]$$

L_1 = node#1 has two neighbors with 1 rel.

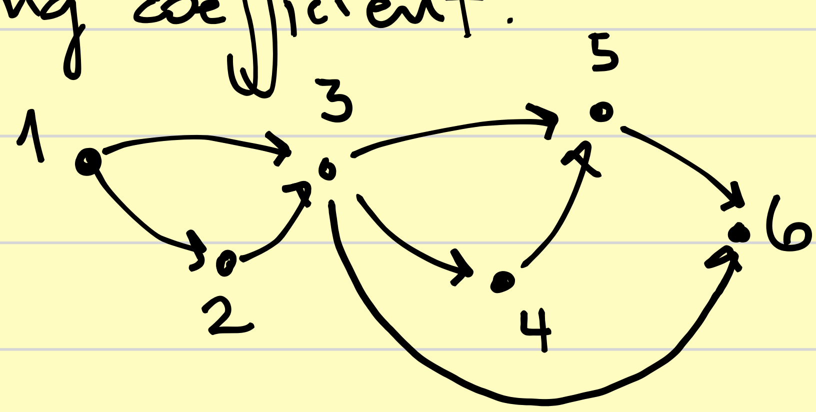


$= \ldots$

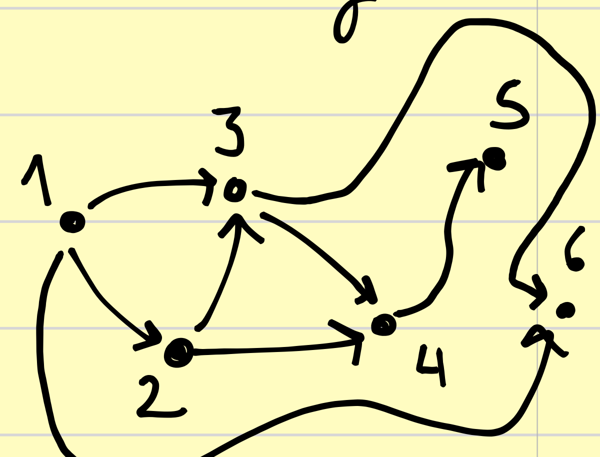
The higher the clustering coefficient, the better the communication between the elements of the network.

Example. Compare these two graphs in terms of clustering coefficient.

G_1



G_2



3. LAPLACIAN NETWORK

Laplacian matrix of a network describes/contains all relevant information of a network.

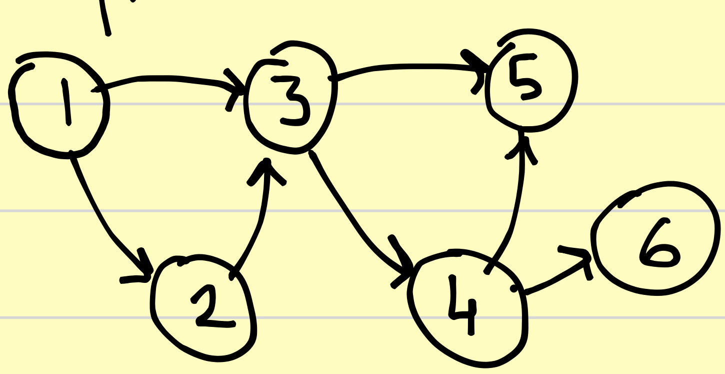
Is defined as: $\boxed{L = D - A}$

$L \equiv$ Laplacian Matrix
$D \equiv$ Degree Distribution Mat.
$A \equiv$ Adjacency Matrix

$D \equiv$ degree matrix $\begin{cases} k_i & i=j \\ 0 & i \neq j \end{cases}$ $\quad k_i \equiv$ degree of node

$A \equiv$ adjacency matrix $\begin{cases} 1 & \text{if } i \,\&\, j \text{ have a connection} \\ 0 & \text{otherwise.} \end{cases}$

Example.



$$L = \underbrace{\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 2 & & & & & \\ 2 & & 2 & & \emptyset & & \\ 3 & & & 4 & & & \\ 4 & & & & 3 & & \\ 5 & & \emptyset & & & 2 & \\ 6 & & & & & & 1 \end{array}}_{D} - \underbrace{\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}}_{A} =$$

$$
L = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{c}
\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{bmatrix}
2 & -1 & -1 & 0 & 0 & 0 \\
0 & 2 & -1 & 0 & 0 & 0 \\
0 & 0 & 4 & -1 & -1 & 0 \\
0 & 0 & 0 & 3 & -1 & -1 \\
0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & & 1
\end{bmatrix}
\end{array}
$$

NETWORK $\longrightarrow$ LAPLACIAN MATRIX

$G = \{N, E\}$ $\quad$ L

Non euclidean $\quad$ Euclidean

- The eigenvectors of the laplacian matrix contain the spectral information of the network.

$$\boxed{A\vec{v} = \lambda\vec{v} \longrightarrow \det|A - \lambda I| = 0 \longrightarrow \lambda \longrightarrow \vec{v}}$$

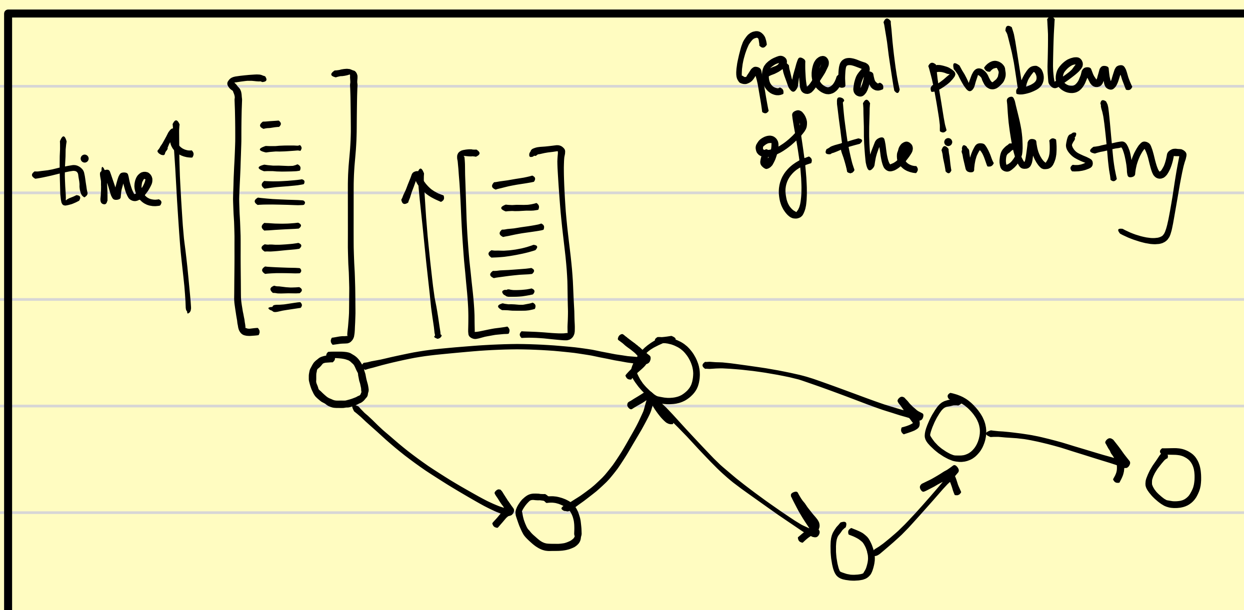
- The second eigenvector of the laplacian matrix of the graph is called the Fiedler vector.
- The Fiedler vector has an important property: it has a sign structure that shows the bottleneck of the graph.

$$
G(N,E) \longrightarrow L \longrightarrow \underset{\text{Fiedler vector}}{\vec{v}_2} =
\begin{bmatrix} + \\ + \\ + \\ 0 \\ 0 \\ \vdots \\ 0 \\ - \\ - \end{bmatrix}
\begin{array}{c} 1 \\ 2 \\ 3 \\ \\ \vdots \\ \\ \\ \\ N \end{array}
$$

$\leftarrow$ Bottleneck of the graph

Excercise. please write a Python code 3.x that generates two graphs, each with 100 nodes, and each node attached to a vector of 300 elements that is normally distributed $N_1(k_i, \sqrt{k_i})$; and
NETWORK 1
$N_2(k_i^2, \sqrt[3]{k_i})$.
NETWORK 2

The code should then calculate the fiedler vector of each graph and you should interpret the results. The networks should have a Barabasi structure with $\gamma = 2'4$ (exponent degree).



NETWORK TOPOLOGY

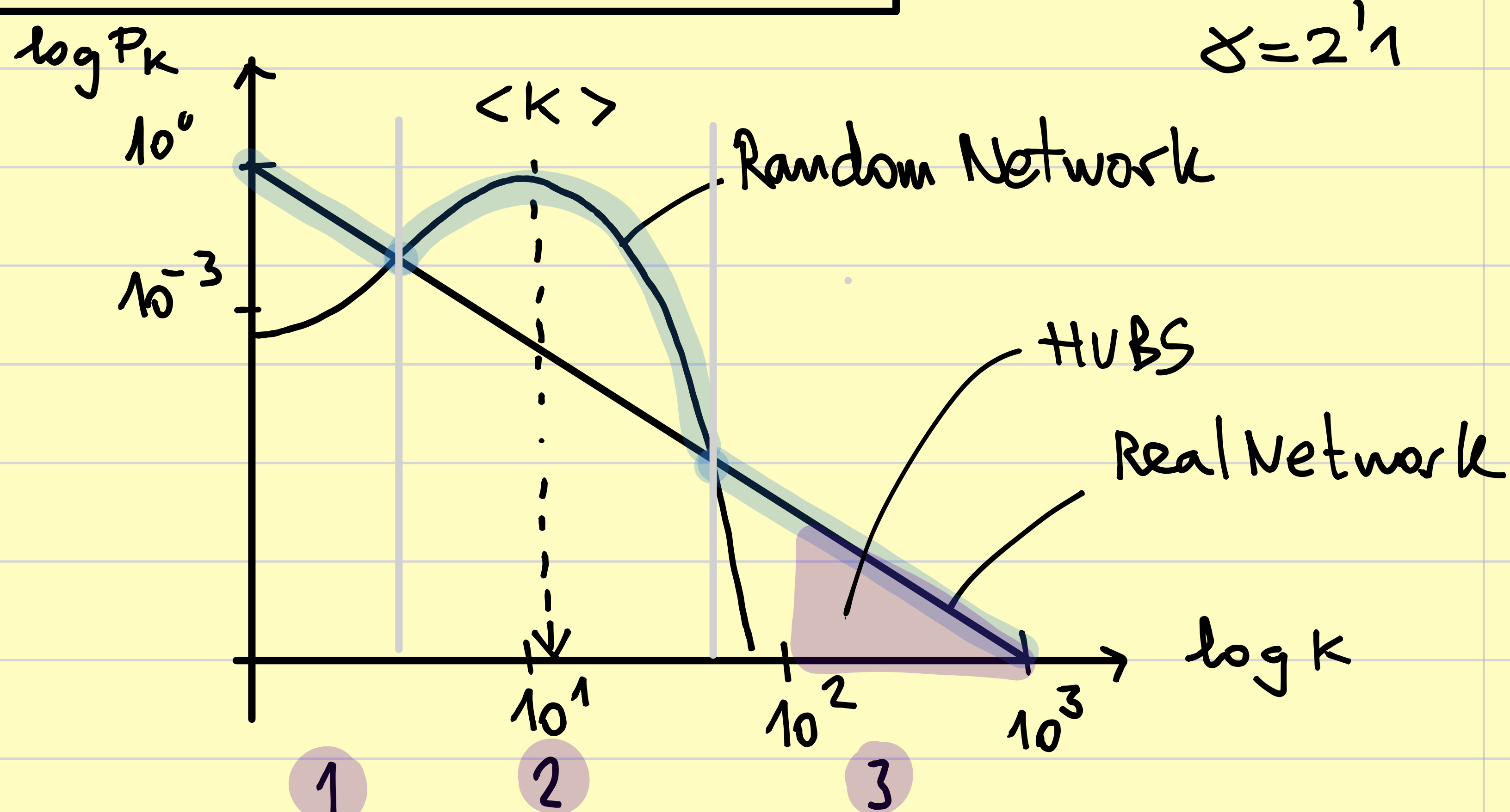
- The probability that a node with k neighbours attaches a new node in a random network is given by a Poisson distribution with parameter λ:

$$\boxed{P_k \equiv P_{\text{Random}}(X=k) = \frac{\lambda^k e^{-\lambda}}{k!}}$$

• The probability that a node is connected to another with "k" neighbours in real networks is given by a POWER LAW distribution:

$$\boxed{P_K \equiv P_{Real}(X=k) = k^{-\gamma} \; ; \quad \gamma \equiv \text{Exponent degree}}$$

NETWORK SCIENCE ; Baràbasi 2016



1 RANDOM << REAL. For small K the power law of the real network is above the poisson (random), indicating that the real world has more nodes with few neigbours.

2 RANDOM > REAL. For nodes around the average degree <k>, the random network has excess of nodes.

3 RANDOM << REAL. for large K the real network presents more nodes with high K.

The role of the exponent γ :

$\gamma \leq 2$. Anomalous regime. Networks cannot exist long.

$2 < \gamma < 3$. Scale free regime. " grow healthy

$\gamma > 3$. Random regime.