GRAPH FOR TRANSPORTATION NETWORKS

An introduction

WHO AM I ?

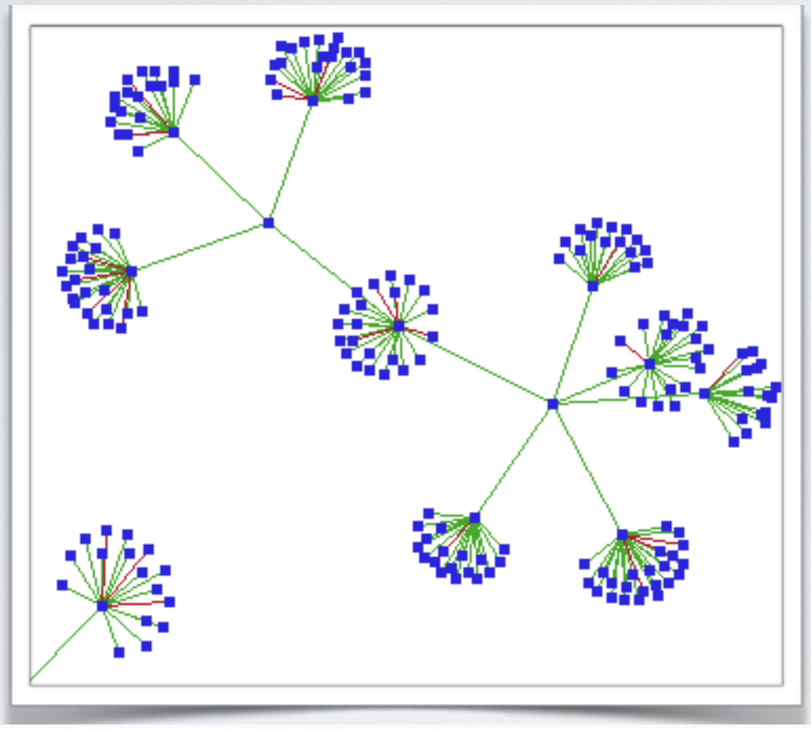
- Associate professor (Maître de conférences) at LIRIS lab, UCBL, Lyon University

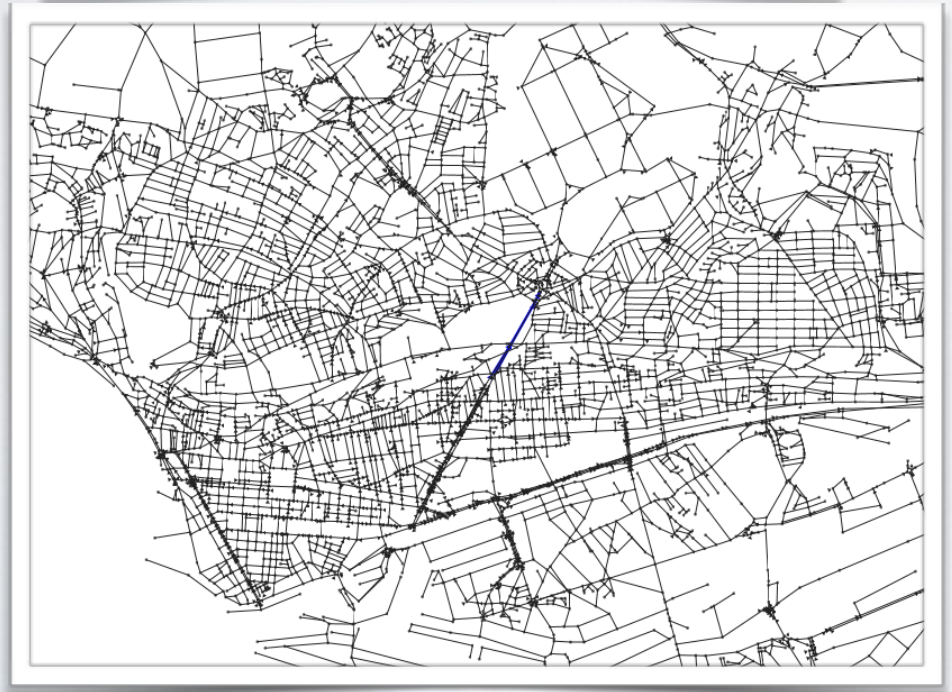
- Working on
  - **Complex Networks**
  - **Complex Systems**
  - **Data Mining/Machine Learning**

OUTLINE

- Introduction
- Analyzing graphs as a whole
- Analyzing nodes and edges
- Complexifying Complex Networks
- Machine Learning with graphs

COMPLEX NETWORKS





COMPLEX NETWORKS

- Difference with “graph theory”:
  - Difference of vocabulary, scientific communities
  - Study real data from the real world
  - Not regular (grid…), not Random …
  - Not the same questions (node coloring, counting cliques, complexity…)
- What about “networks” ?
  - Maybe to avoid confusion with the computer science term…

COMPLEX NETWORKS

- Complex networks can be:
  - Directed (one-directional edge)
  - Weighted (nodes/edges have different weights)
  - Attributed (nodes/edges have properties)
  - Dynamic (nodes/edges/properties change)
  - Spatial (nodes/edges have locations in a space)
  - Multigraph (multiple edges between same nodes)
  - Multiplex (Edges different nature, i.e. several networks with same nodes)
  - Multipartite (Nodes of different nature, e.g. products and individuals)
  - …
- Think of the representation of a street network…

COMPLEX NETWORKS

- Big questions:
  - Important nodes ? (centrality)
  - Important edges ?
  - Diffusion/Flows on the network ?
  - Organisation of the network (clusters, core-periphery…) ?
  - Resilience (robust to default)
  - Micro/Meso/Macro properties (clustering, degree distribution…)
  - …

COMPLEX NETWORKS

- Machine learning on graphs
  - Mainly using **graph embedding**
  - Machine learning usually => predict properties from properties
  - On graph: predict structure from properties, properties from structure, etc.

- Machine Learning (AI) tasks:
  - Classification of nodes or edges
    - Supervised, unsupervised
  - Prediction of values of nodes or edges
  - Prediction of properties of incoming nodes
  - …

COMPLEX SYSTEMS

- Examples of complex systems:
  - Societies
  - Economies
  - Human body
  - Brain
  - Ecosystems
  - **Cities**
  - **Public Transportation Systems**
  - **Traffic**
  - etc.

COMPLEX SYSTEMS

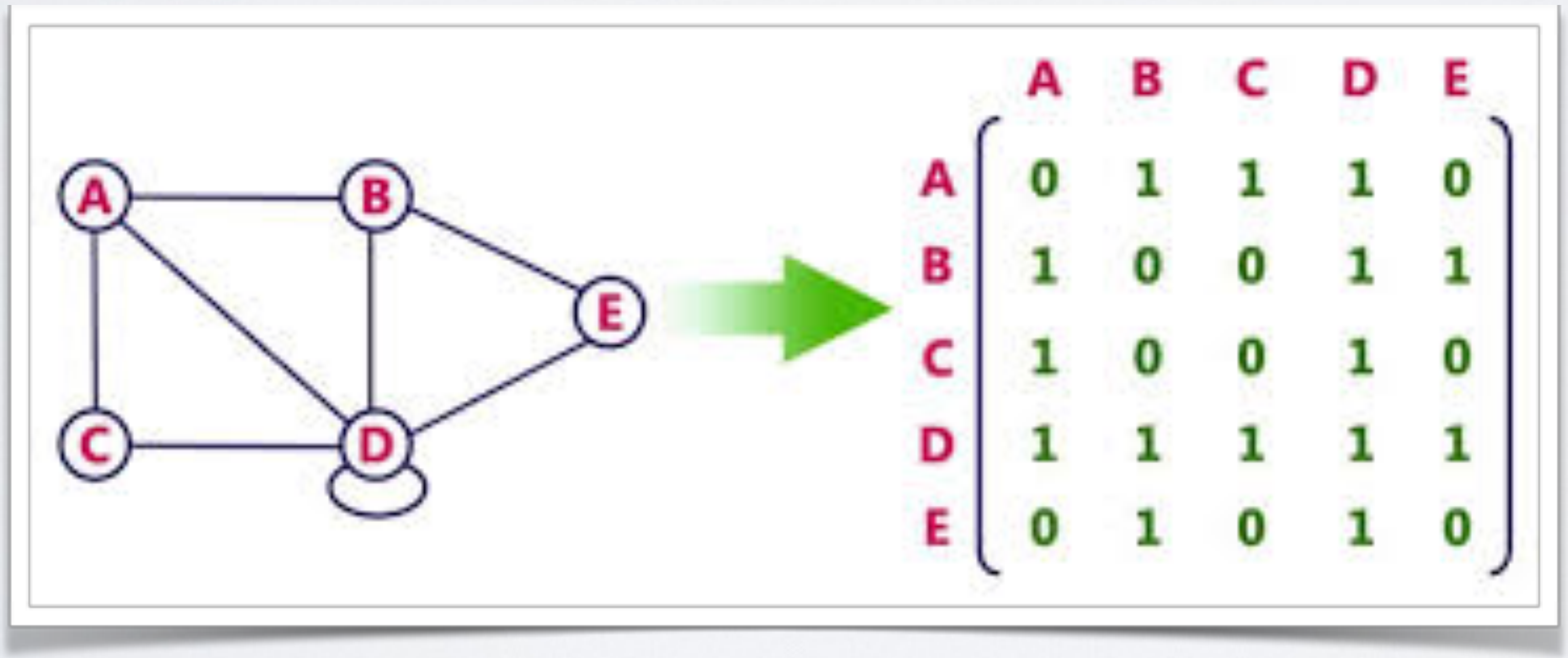
- Complex systems often have common phenomenons:
  - *Emergence* of macro phenomenons
  - Chaotic behaviour ("Butterfly effect")
  - Modular organisation
  - Multi-Scale organisation
  - "phase transition" or "percolation effect"
  - …

- Think of traffic jams, trip patterns (holidays, commute), incidents impact on the system, …

GRAPHS: INTRODUCTION

GRAPHS ?

- A graph can be represented as:
  - A list of edges : [{v1,v2}, {v1,v3}, {v5,v7},…]
  - A neighborhood list: {v1:{v2,v3},v2:{v1},v5:{v7},…}
  - An adjacency matrix



$$
\begin{array}{c|ccccc}
 & A & B & C & D & E \\
\hline
A & 0 & 1 & 1 & 1 & 0 \\
B & 1 & 0 & 0 & 1 & 1 \\
C & 1 & 0 & 0 & 1 & 0 \\
D & 1 & 1 & 1 & 1 & 1 \\
E & 0 & 1 & 0 & 1 & 0
\end{array}
$$

SIZE

- A network is composed of nodes and edges.
- Size: How many nodes and edges ?

	#nodes	#edges
Wikipedia HL	2M	30M
Twitter 2015	288M	60B
Facebook 2015	1.4B	400B
Brain c. Elegans	280	6393
Roads US	2M	2.7M
Airport traffic	3k	31k

DENSITY

Defined as:

Directed

$$D = \frac{|E|}{|V|\,(|V|-1)}$$

Undirected

$$D = \frac{2|E|}{|V|\,(|V|-1)}$$

Often more relevant: average degree (2|E| / |V|)

	#nodes	#edges	Density	avg. deg
Wikipedia	2M	30M	1.5×10^{-5}	30
Twitter 2015	288M	60B	1.4×10^{-6}	416
Facebook	1.4B	400B	4×10^{-9}	570
Brain c.	280	6393	0.16	46
Roads Calif.	2M	2.7M	6×10^{-7}	2.7
Airport	3k	31k	0.007	21

DEGREE DISTRIBUTION

- In a fully random graph (Erdos-Renyi), degree distribution is a normal distribution centered on the average degree
- In real graphs, in general, it is not the case:
  - A high majority of small degree nodes
  - A small minority of nodes with very high degree (Hubs)
- Often modeled by a **power law**

DEGREE DISTRIBUTION

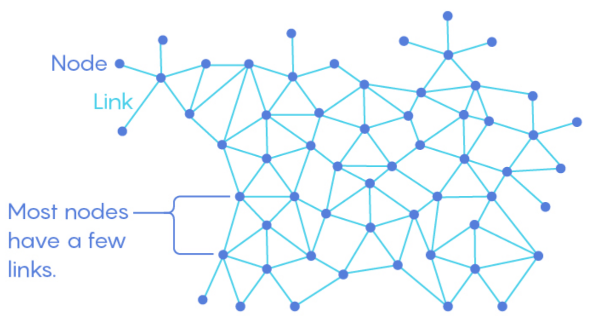
Power law/Scale free distribution: $f(x) = ax^{-k}$

To Be or Not to Be Scale-Free

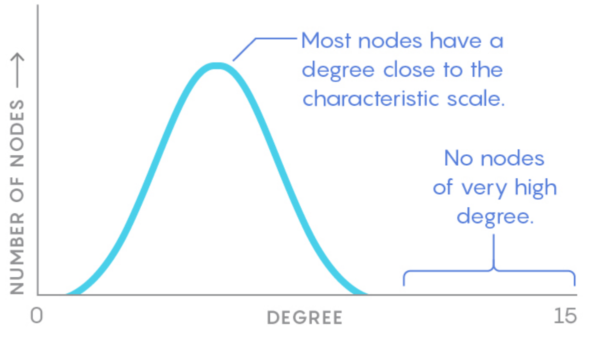
Scientists study complex networks by looking at the distribution of the number of links (or "degree") of each node.
Some experts see so-called scale-free networks everywhere. But a new study suggests greater diversity in real-world networks.

Random Network

Randomly connected networks have nodes with similar degrees. There are no (or virtually no) "hubs" — nodes with many times the average number of links.

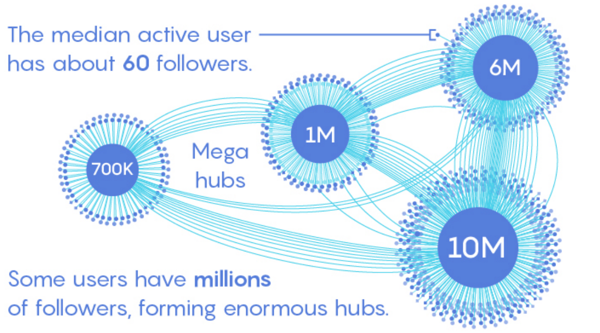


The distribution of degrees is shaped roughly like a bell curve that peaks at the network's "characteristic scale."

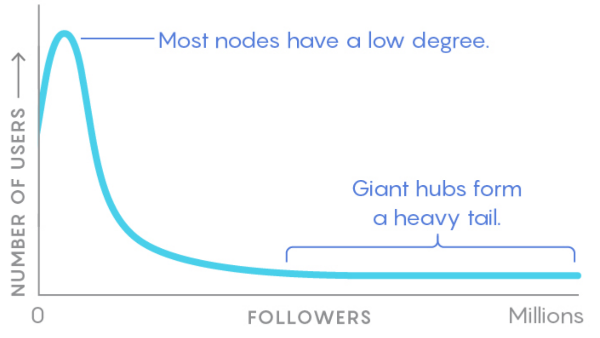


Twitter's Scale-Free Network

Most real-world networks of interest are not random. Some nonrandom networks have massive hubs with vastly higher degrees than other nodes.

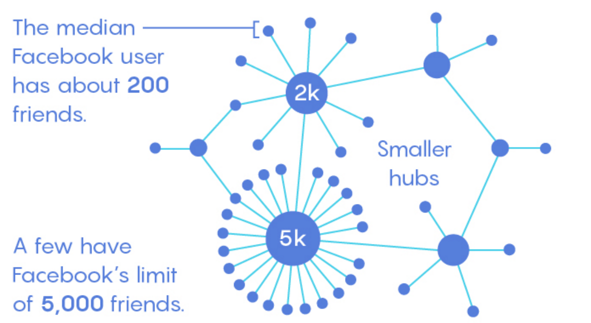

The median active user has about 60 followers.
6M
1M
700K
Mega hubs
10M
Some users have millions of followers, forming enormous hubs.

The degrees roughly follow a power law distribution that has a "heavy tail." The distribution has no characteristic scale, making it scale-free.

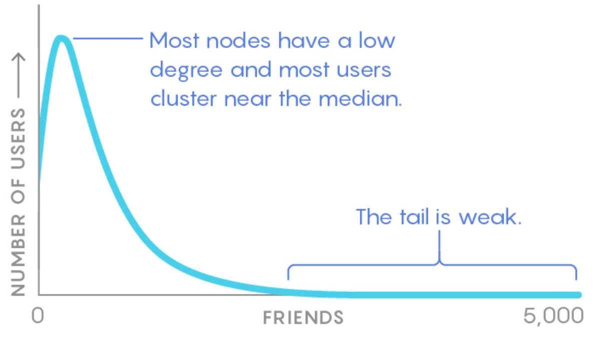


Facebook's In-Between Network

Researchers have found that most nonrandom networks are not strictly scale-free. Many have a weak heavy tail and a rough characteristic scale.


The median Facebook user has about 200 friends.
2k
Smaller hubs
5k
A few have Facebook's limit of 5,000 friends.

This network has fewer and smaller hubs than in a scale-free network. The distribution of nodes has a scale and does not follow a pure power law.



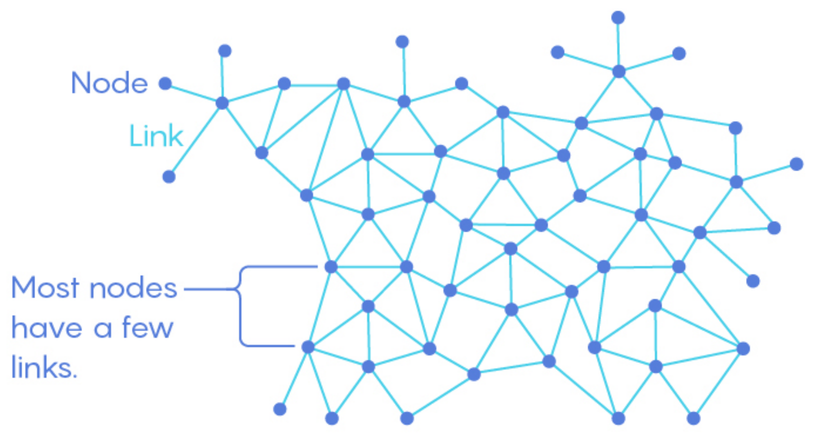
[Quanta magazine 2018]

To Be or Not to Be Scale-Free

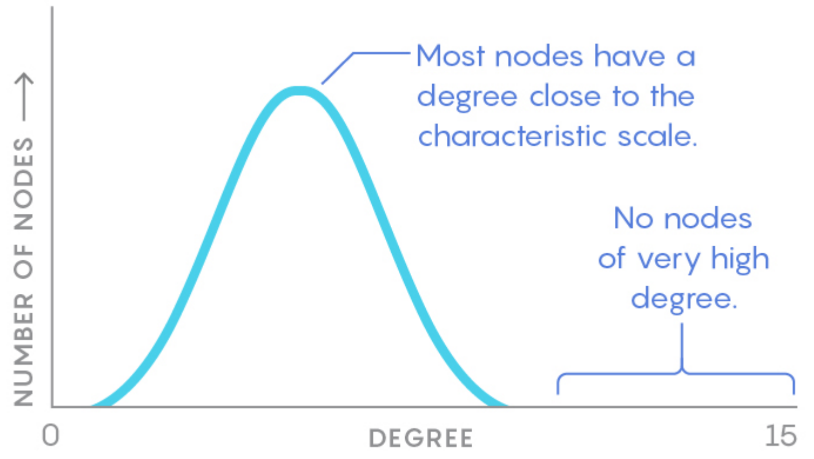
Scientists study complex networks by looking at the distribution of the number of links (or "degree") of each node. Some experts see so-called scale-free networks everywhere. But a new study suggests greater diversity in real-world networks.

Random Network

Randomly connected networks have nodes with similar degrees. There are no (or virtually no) "hubs" — nodes with many times the average number of links.

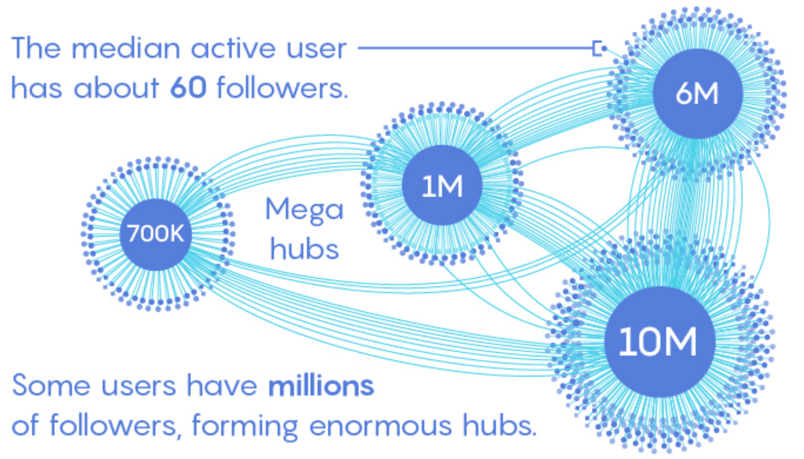


The distribution of degrees is shaped roughly like a bell curve that peaks at the network's "characteristic scale."



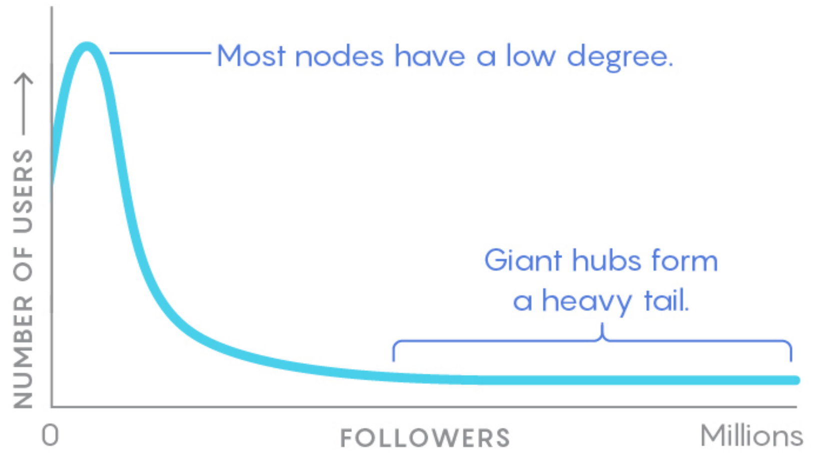
Twitter's Scale-Free Network

Most real-world networks of interest are not random. Some nonrandom networks have massive hubs with vastly higher degrees than other nodes.



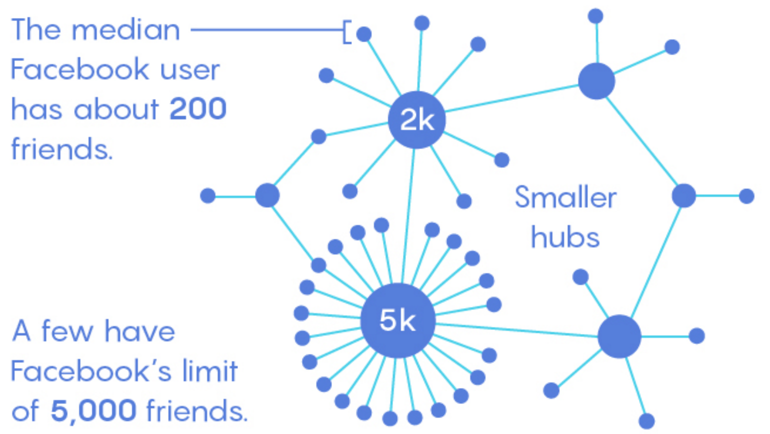
The median active user has about **60** followers.
6M
1M
700K
Mega hubs
10M
Some users have **millions** of followers, forming enormous hubs.

The degrees roughly follow a power law distribution that has a "heavy tail." The distribution has no characteristic scale, making it scale-free.



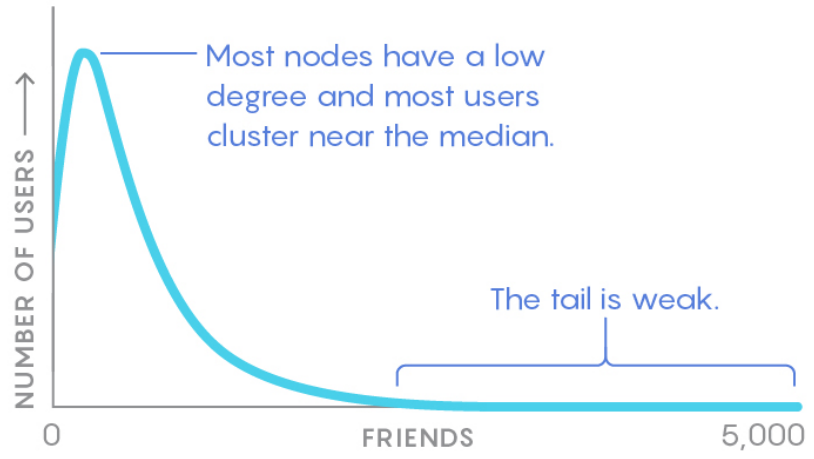
Facebook's In-Between Network

Researchers have found that most nonrandom networks are not strictly scale-free. Many have a weak heavy tail and a rough characteristic scale.



The median Facebook user has about **200** friends.
2k
Smaller hubs
5k
A few have Facebook's limit of **5,000** friends.

This network has fewer and smaller hubs than in a scale-free network. The distribution of nodes has a scale and does not follow a pure power law.



CLUSTERING COEFFICIENT

Global clustering coefficient

$$C = \frac{\text{number of closed triplets}}{\text{number of all triplets (open and closed)}}.$$

Triplet: set of 3 nodes connected by 2 or 3 edges

Average Clustering Coefficient

Clustering coefficient of a node: $C_i = \frac{2|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$

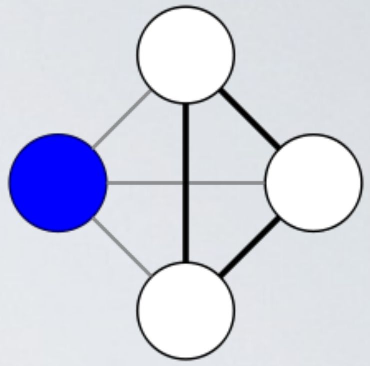
Average CC: $\bar{C} = \frac{1}{n}\sum_{i=1}^{n} C_i$

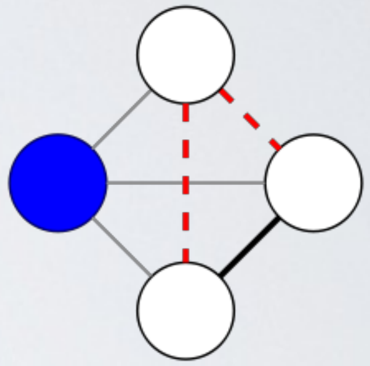
CLUSTERING COEFFICIENT

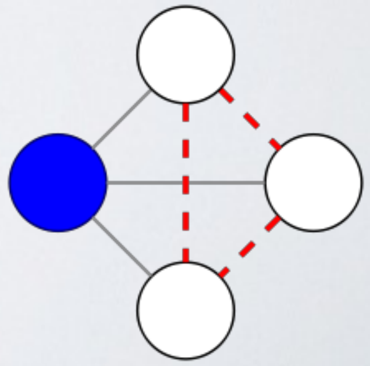
The higher the value,
the more **locally dense** is the network.

"Friends of my friends are my friends"

Higher in <u>real networks</u> than random

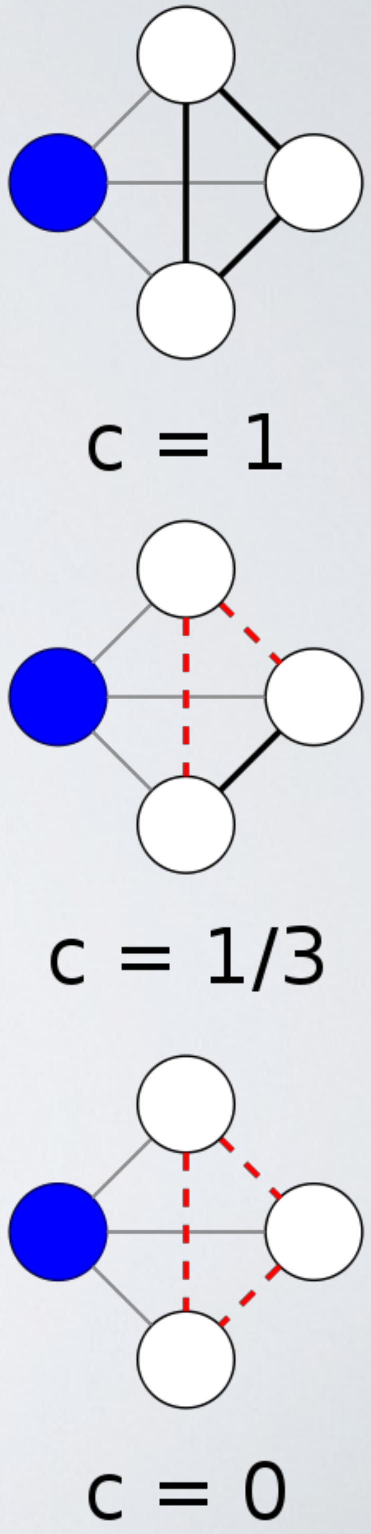

c = 1


c = 1/3


c = 0

CLUSTERING COEFFICIENT

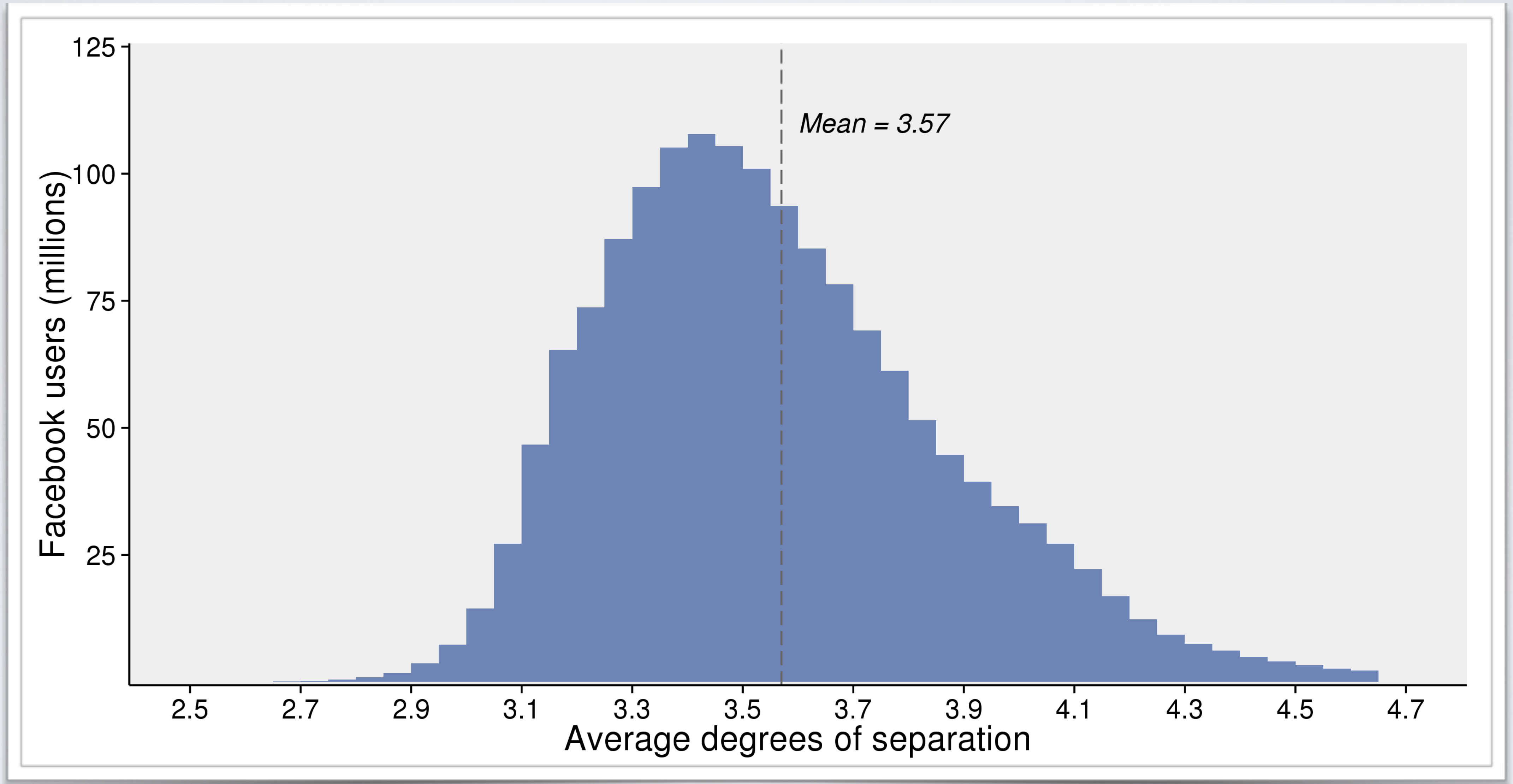
- Facebook ego-networks: 0.6
- Twitter lists: 0.56
- California Road networks: 0.04
- Random (ER): =density: very small for large graphs



AVERAGE PATH LENGTH

- Average shortest path between all pairs of nodes
- The famous 6 degrees of separation (Milgram experiment)
  - In fact 6 hops
  - (More on that next slide)
- Not too sensible to noise
- Tells you if the network is “stretched” or “hairball” like

SIDE-STORY: MILGRAM EXPERIMENT



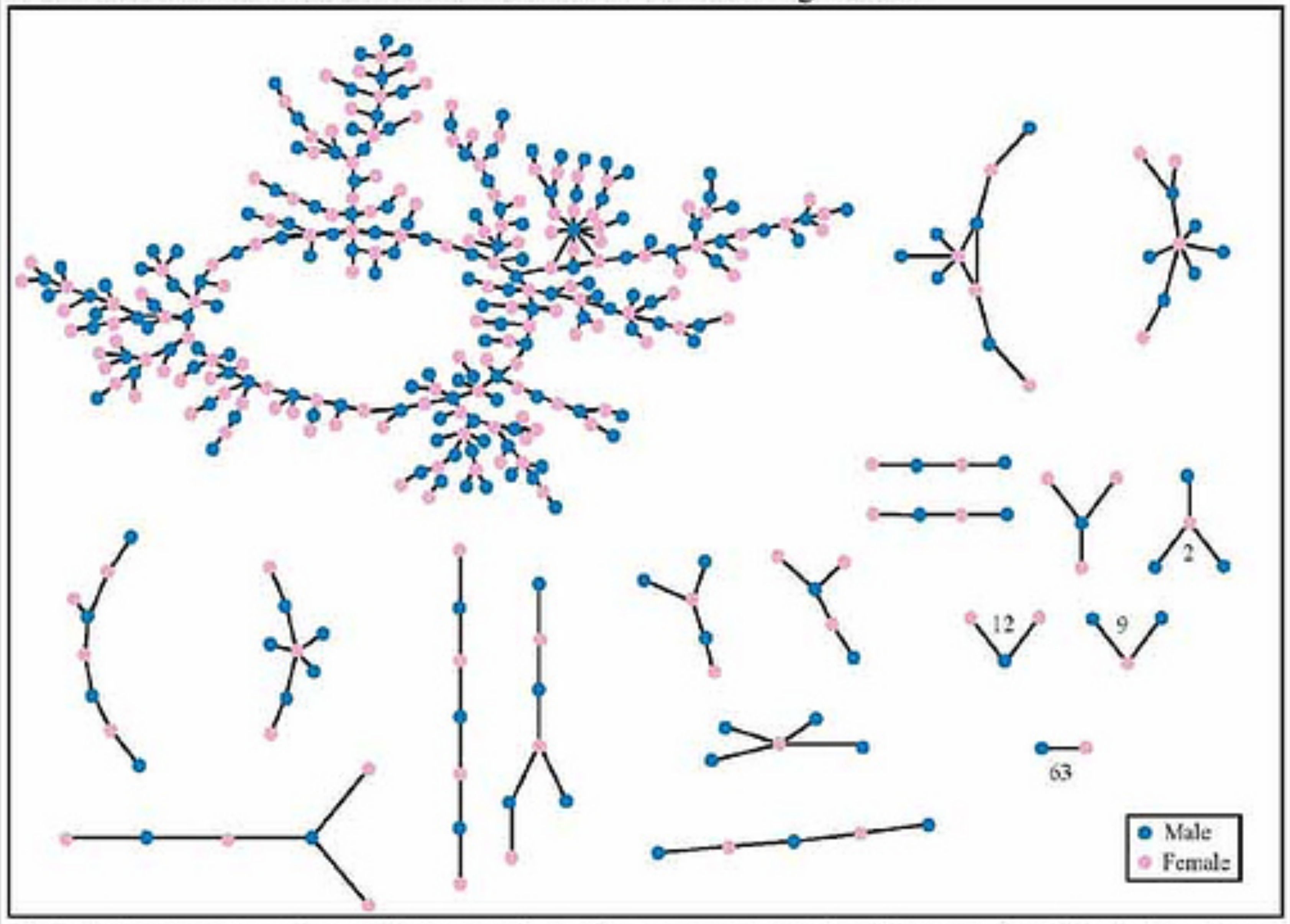
Facebook

HOMOPHILY/ASSORTATIVITY

- Nodes might have a preference for some other nodes
  - Similar nodes (age in social networks)
  - Different nodes (genre in sentimental networks (yes, it has been done!))
  - Nodes with a particular property
- "Assortativity" alone often used to mean "degree assortativity"
  - Large nodes are preferentially connected to large nodes
- All this implies: "compared with a random network"

HOMOPHILY/ASSORTATIVITY

The Structure of Romantic and Sexual Relations at "Jefferson High School"



Each circle represents a student and lines connecting students represent romantic relations occuring within the 6 months preceding the interview. Numbers under the figure count the number of times that pattern was observed (i.e. we found 63 pairs unconnected to anyone else).

HOMOPHILY/ASSORTATIVITY

- Nodes might have a preference for some other nodes
  - Similar nodes (age in social networks)
  - Different nodes (genre in sentimental networks (yes, it has been done!))
  - Nodes with a particular property
- "Assortativity" alone often used to mean "degree assortativity"
  - Large nodes are preferentially connected to large nodes
- All this implies: "compared with a random network"

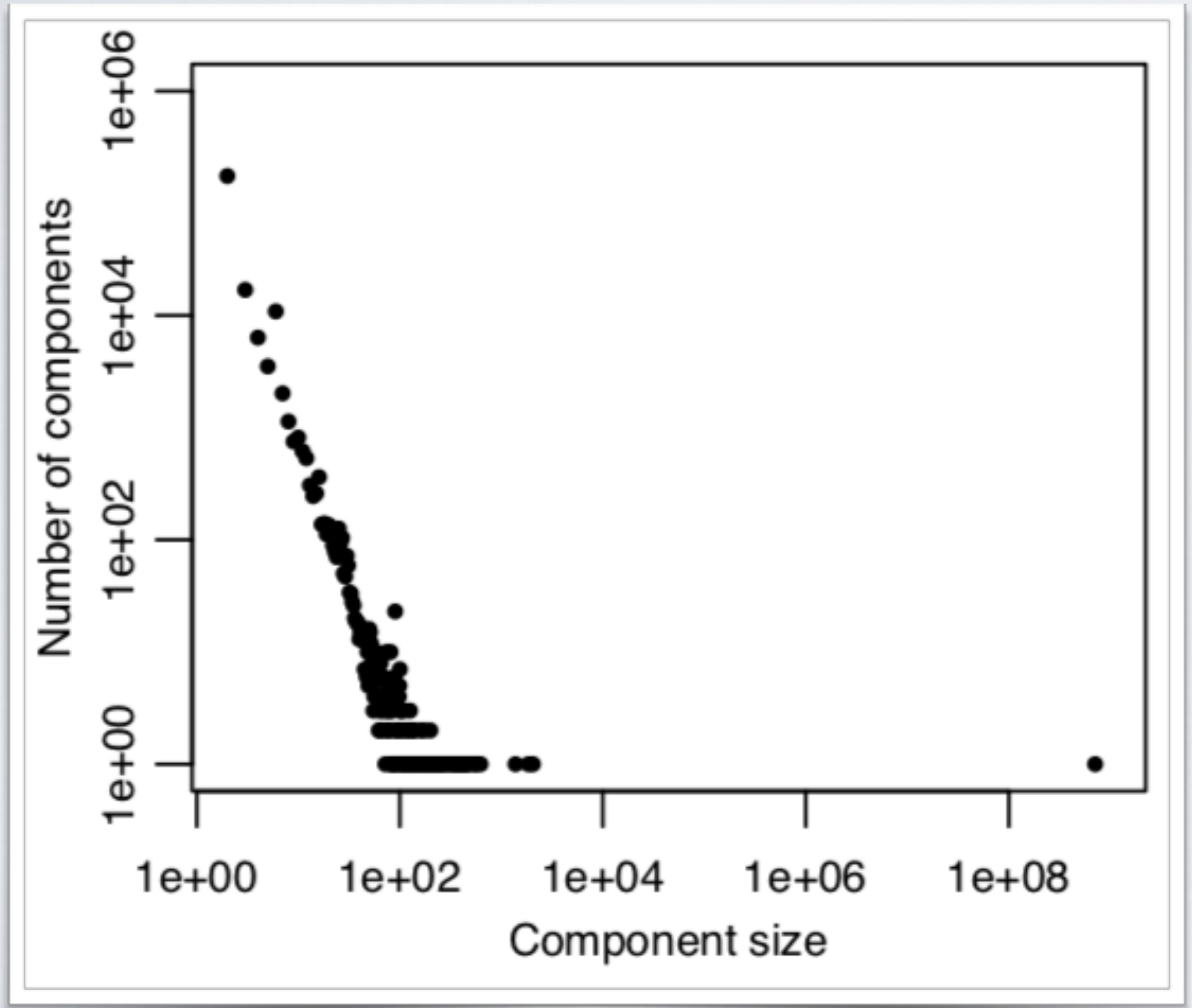
EXEMPLE OF GRAPH ANALYSIS

- Source: [The Anatomy of the Facebook Social Graph, Ugander et al. 2011]
- The Facebook friendship network in 2011

EXEMPLE OF GRAPH ANALYSIS

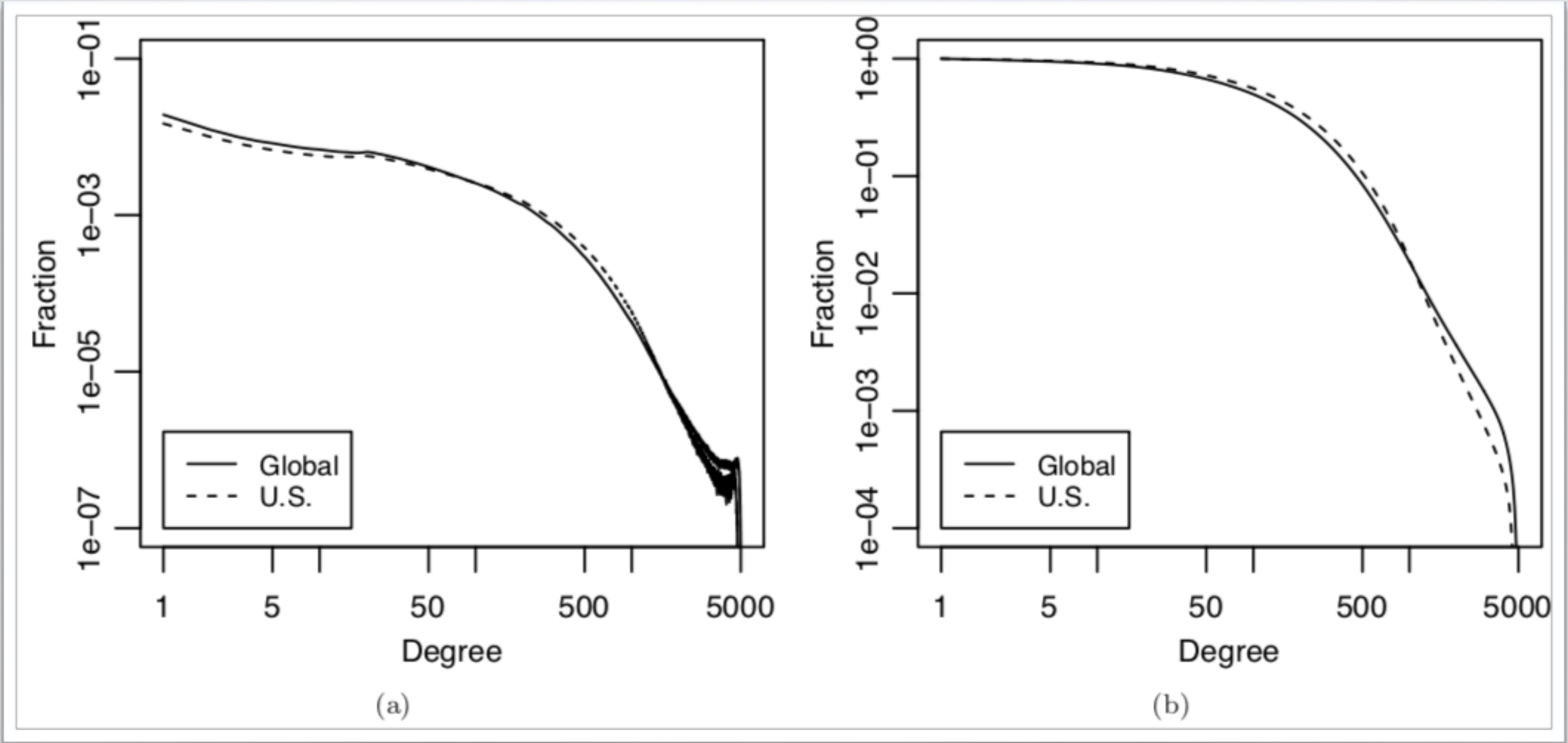
- 721M users (nodes) (active in the last 28 days)
- 68B edges
- Average degree: 190 (average # friends)
- Median degree: 99
- Connected component: 99.91%

EXEMPLE OF GRAPH ANALYSIS



Component size
Distribution

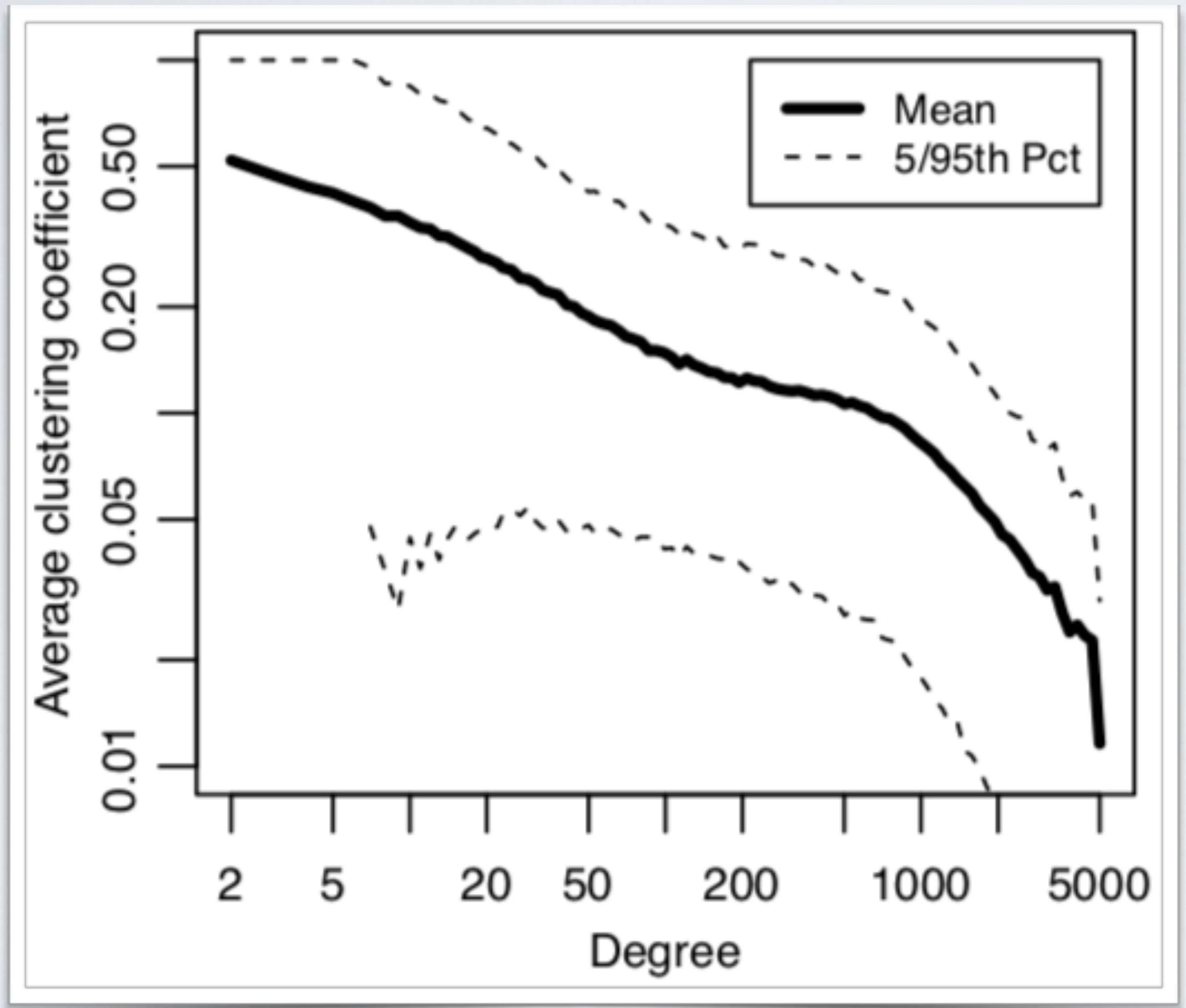
EXEMPLE OF GRAPH ANALYSIS



Cumulative

Degree distribution

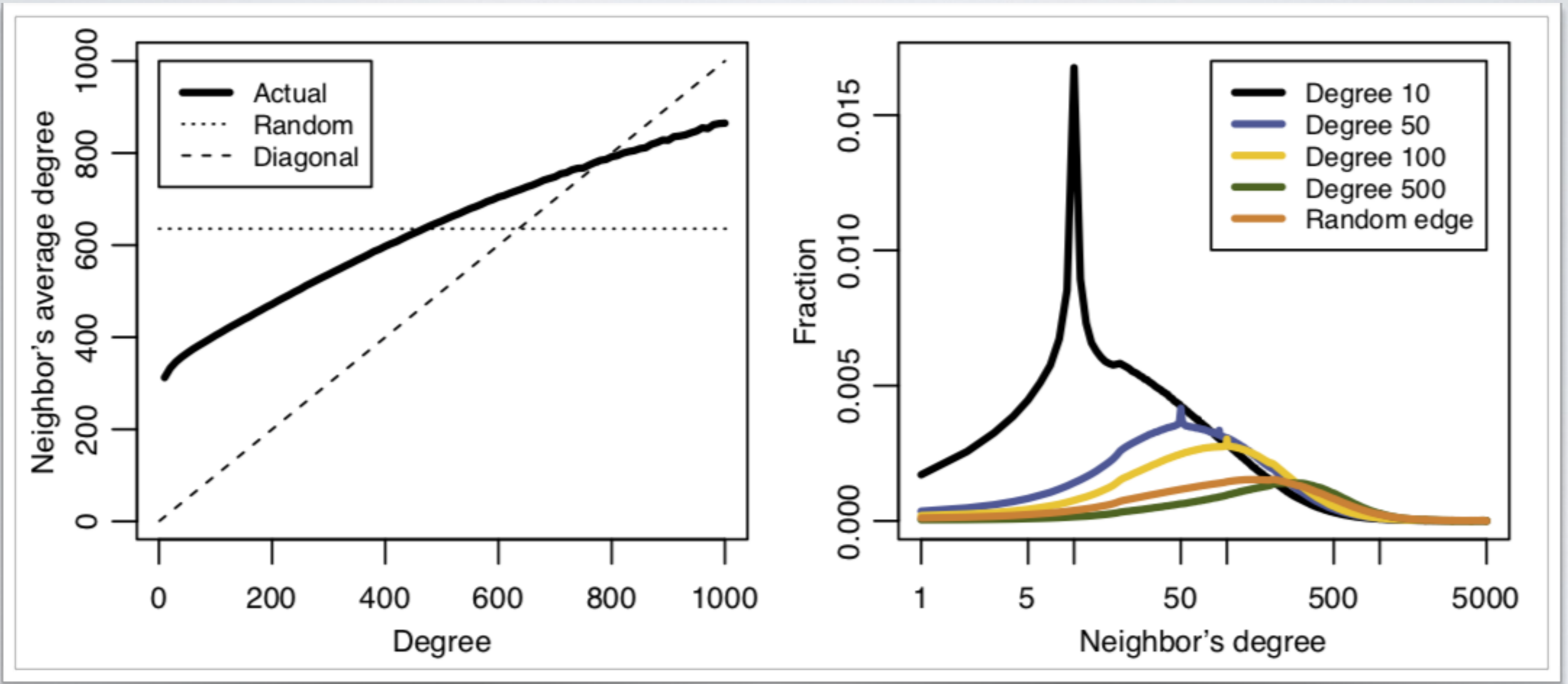
EXEMPLE OF GRAPH ANALYSIS



Clustering coefficient
By degree

Median user: 0.14:
14% of pair friends
Are actually friends

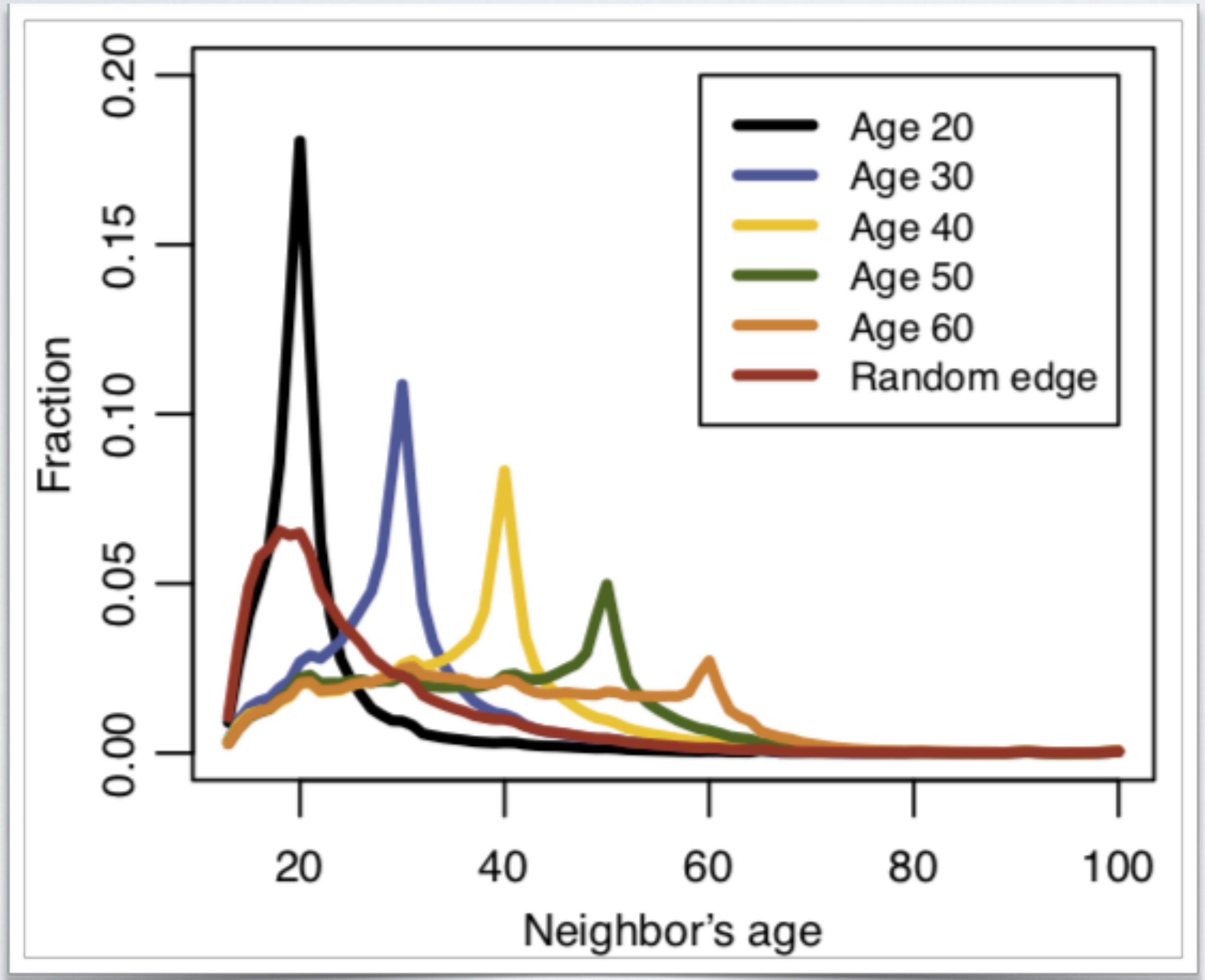
EXEMPLE OF GRAPH ANALYSIS



My friends have more Friends than me!

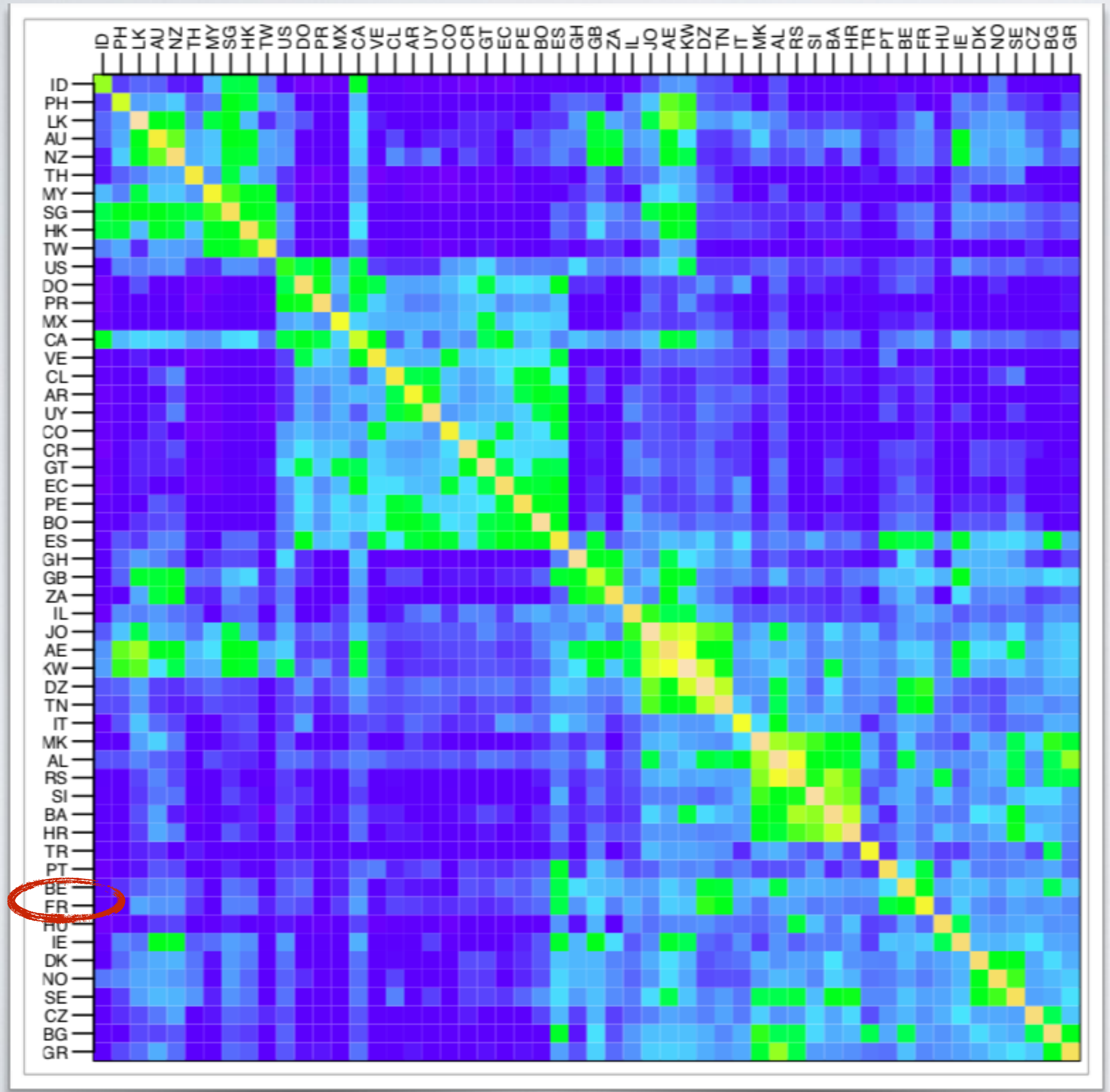
Many of my friends have the Same # of friends than me!

EXEMPLE OF GRAPH ANALYSIS



Age homophily

EXEMPLE OF GRAPH ANALYSIS



Country similarity

84.2% percent of edges are within countries

(See community detection)

DESCRIPTION AT THE MICRO-LEVEL

MACRO-MICRO

- First part: description of the graph at the macro level
- Second part:
  - How to describe each element ?
  - How to find "exceptional" elements ?

NODE

- We can measure nodes importance using so-called **centrality**.
- Bad term: nothing to do with being central in general
- Common practice: run many centralities and check relation between centralities and properties/identity of nodes

NODE DEGREE

- **Degree**: how many neighbors

- Often enough to find important nodes
  - Main characters of a series talk with the more people
  - Largest airports have the most connections
  - …

- But not always
  - Facebook users with the most friends are spam
  - Webpages/wikipedia pages with most links are simple lists of references
  - …

NODE DEGREE

- In directed networks, degree is split in:
  - In-degree
  - Out-degree

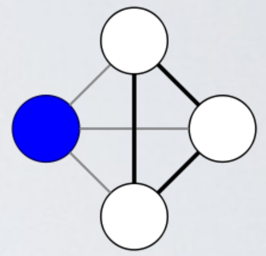
- Example: web pages:
  - Highest out-degree: list of references
  - Highest in-degree: website that attracts a lot of link: probably interesting

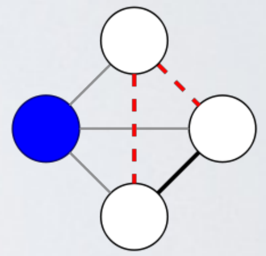
NODE STRENGTH

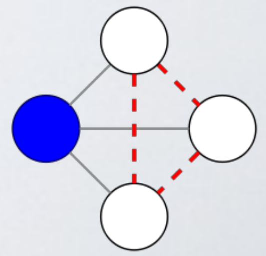
- **Strength**: Degree in a weighted network
- Sum of the weight of adjacent edges

NODE CLUSTERING COEFFICIENT

- **Clustering coefficient**: already seen for global analysis
- The local version
- Tells you if the neighbors of the node are connected
- Be careful!
  - Degree 2: value 0 or 1
  - Degree 1000: Not 0 or 1 (usually)
  - Ranking them is not meaningful


c = 1


c = 1/3

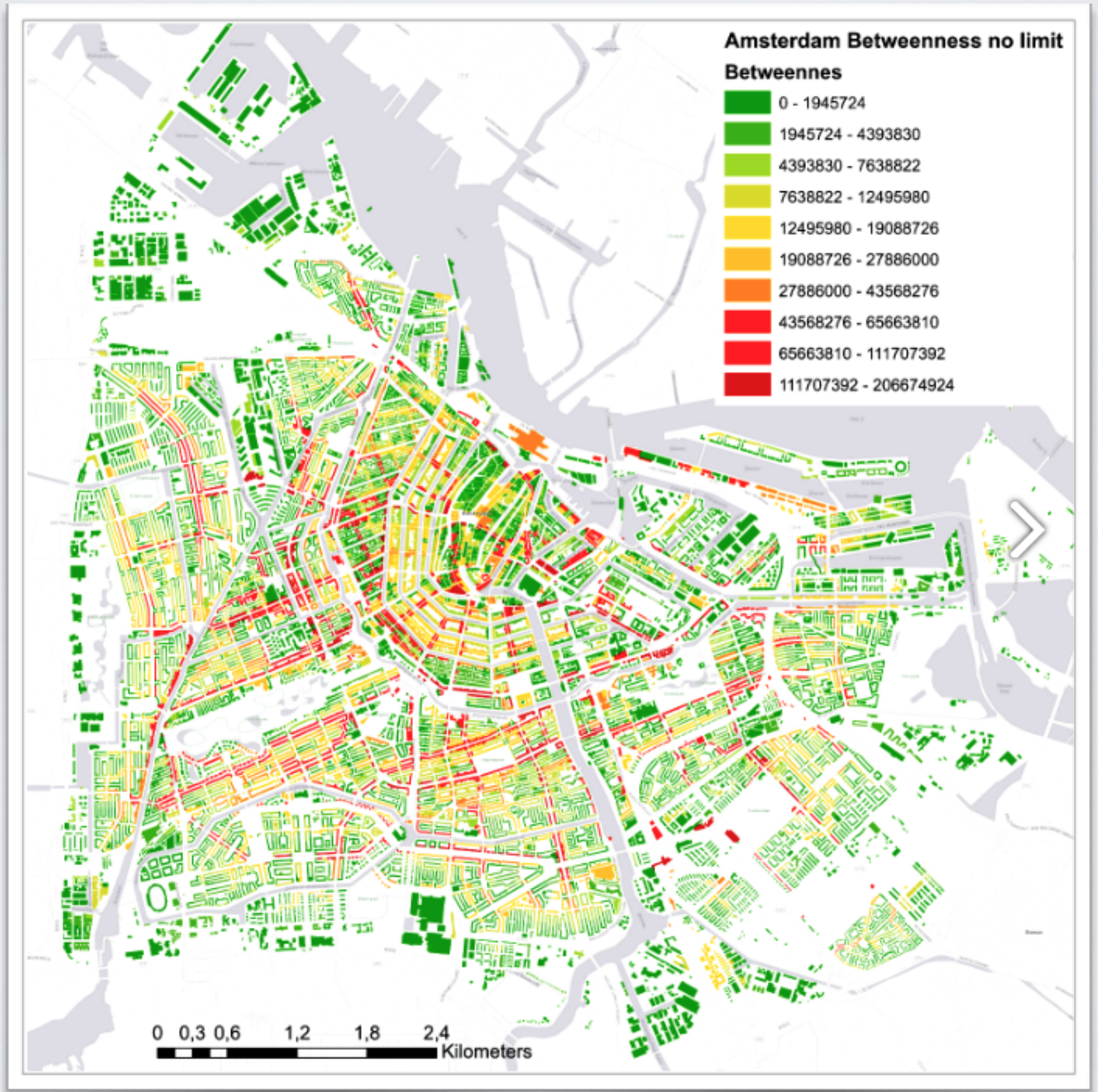

c = 0

NODE BETWEENNESS

- **Betweenness** centrality:
  - 1)compute all shortest paths between all nodes
  - 2)count the fraction of them going through the node
- Idea: if the node is "between" many nodes, then it is important.
- Related to the notion of "flow" of information in the network

NODE BETWEENNESS

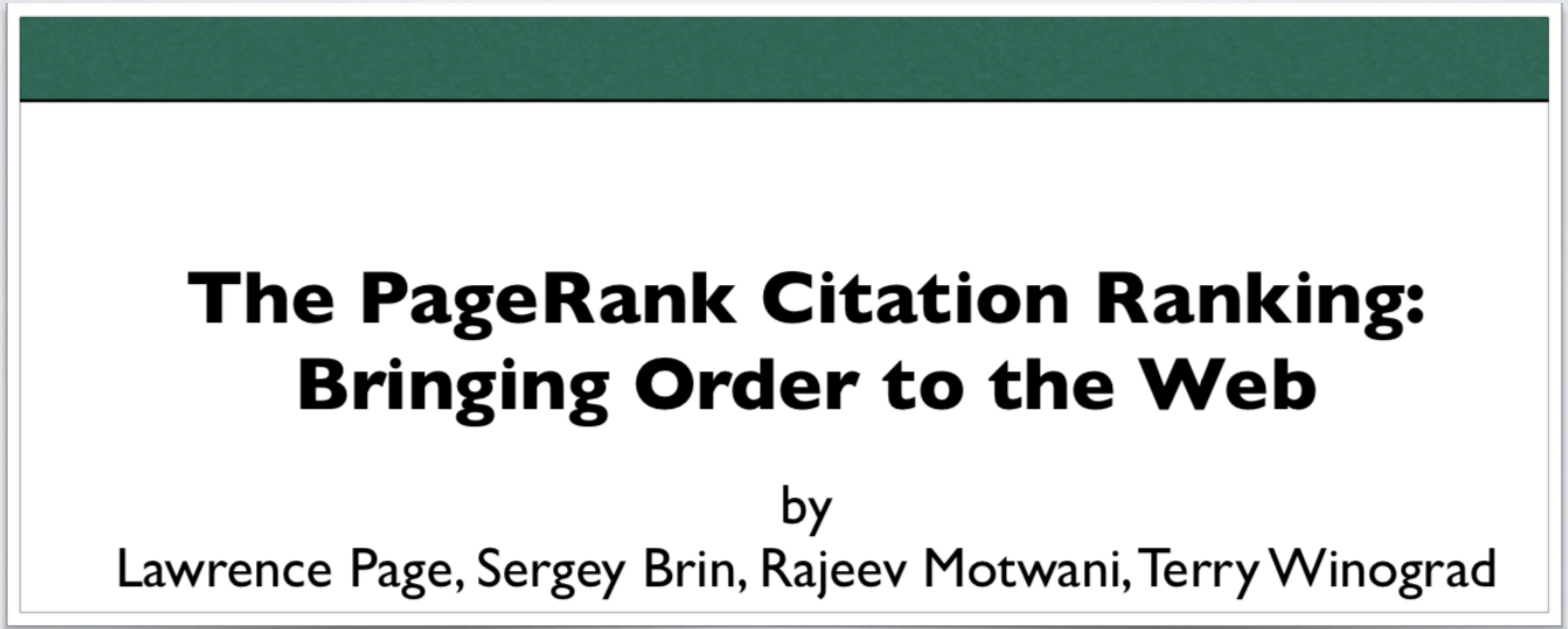
- **Betweenness** centrality:



NODE BETWEENNESS

- **Betweenness** centrality:
- Computationally intractable
- Common approximation:
  - Compute random paths between k nodes (e.g. k=100)

NODE PAGERANK



The PageRank Citation Ranking: Bringing Order to the Web

by

Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd

NODE PAGERANK

- Idea: ranking webpages by relevance
- Problems with in-degree:
  - Easy to fool
  - Where the link comes from Is ignored

NODE PAGERANK

- Solution: Give a score of "authority" to each node determining the score of other nodes
- Interpretation:
  - Likelihood to reach a particular page by clicking links at random
- Parameter:
  - Probability of random hop anywhere to avoid dead end biases
- Computation:
  - Principal eigenvector of the normalized link matrix (including random hops)
  - Power method: random walks on the graph

NODE PAGERANK

- Interpretation: A node is important if many important nodes are linking to it.
- Often correlated with in-degree
- Allow to find tops in hierarchical structures:
    - Commoners talk to local deputy, deputy talk to ministers, ministers talk to the president: the president has low in-degree, but high pagerank.

NODE PAGERANK

- Then how do Google rank when we do a research?
- Create a subgraph of documents related to our topic
- Compute pagerank
- (Of course now it is much more complex…)

EIGENVECTOR CENTRALITY

- Corresponding value of the eigenvector corresponding to the highest eigenvalue of the adjacency matrix
- Crude version of the PageRank

KATZ CENTRALITY

- Variant of the PageRank & Eigenvector centrality

$$C_{\mathrm{Katz}}(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k (A^k)_{ij}$$

KATZ CENTRALITY

- Variant of the PageRank & Eigenvector centrality

$$C_{\mathrm{Katz}}(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k (A^k)_{ij}$$

Katz centrality of node i=

KATZ CENTRALITY

- Variant of the PageRank & Eigenvector centrality

$$C_{\text{Katz}}(i) = \sum_{k=1}^{\infty}\sum_{j=1}^{n} \alpha^k (A^k)_{ij}$$

Repeat for all distances as long
As possible (convergence)

KATZ CENTRALITY

- Variant of the PageRank & Eigenvector centrality

$$C_{\text{Katz}}(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k (A^k)_{ij}$$

Sum for each node **j**

KATZ CENTRALITY

- Variant of the PageRank & Eigenvector centrality

$$C_{\mathrm{Katz}}(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k (A^k)_{ij}$$

Alpha is a parameter.
Its strength decreases at
each iteration

KATZ CENTRALITY

- Variant of the PageRank & Eigenvector centrality

$$C_{\mathrm{Katz}}(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k (A^k)_{ij}$$

Number of different paths from **I** *to* **j** of length k

KATZ CENTRALITY

- Variant of the PageRank & Eigenvector centrality

$$C_{\mathrm{Katz}}(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k (A^k)_{ij}$$

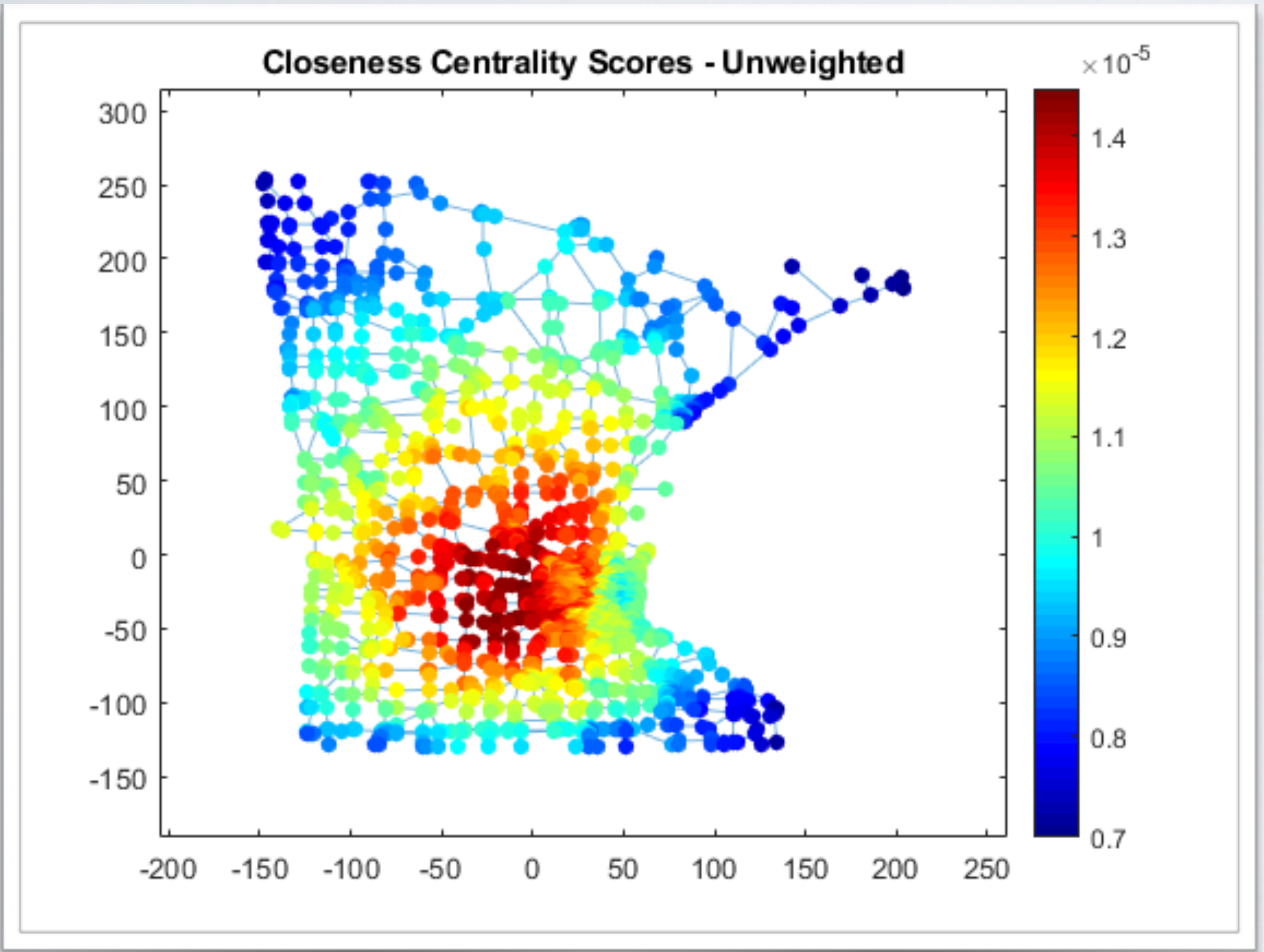
Sum of paths to all other nodes at each distance multiplied by a factor decreasing with distance

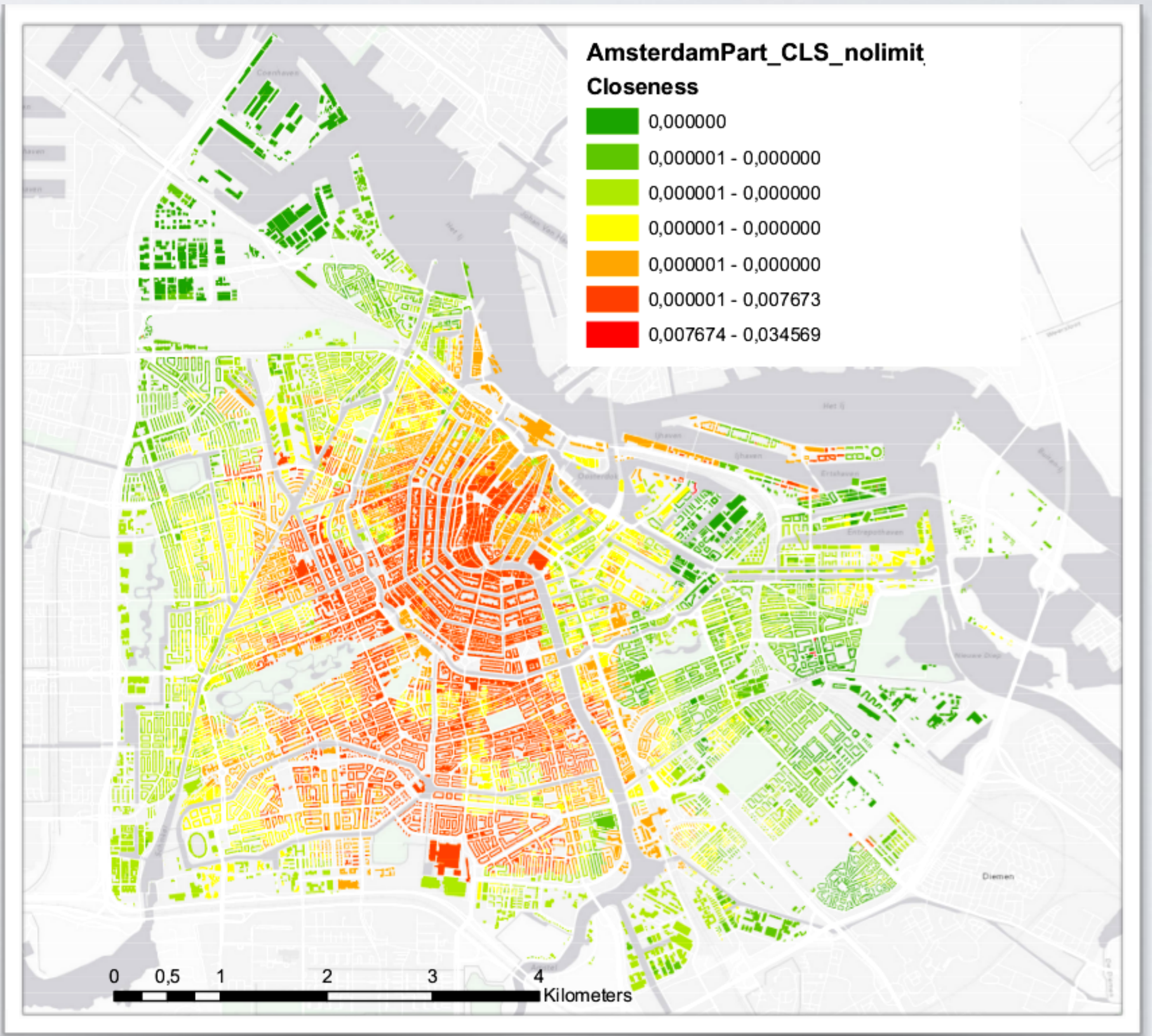
NODE CLOSENESS

$$C(x) = \frac{1}{\sum_y d(y,x)}$$

- **Farness:** sum of length of shortest paths to all other nodes.
- **Closeness**: inverse of the Farness
    - Highest closeness = More central

NODE CLOSENESS





HARMONIC CENTRALITY

$$H(x) = \sum_{y \neq x} \frac{1}{d(y,x)}$$

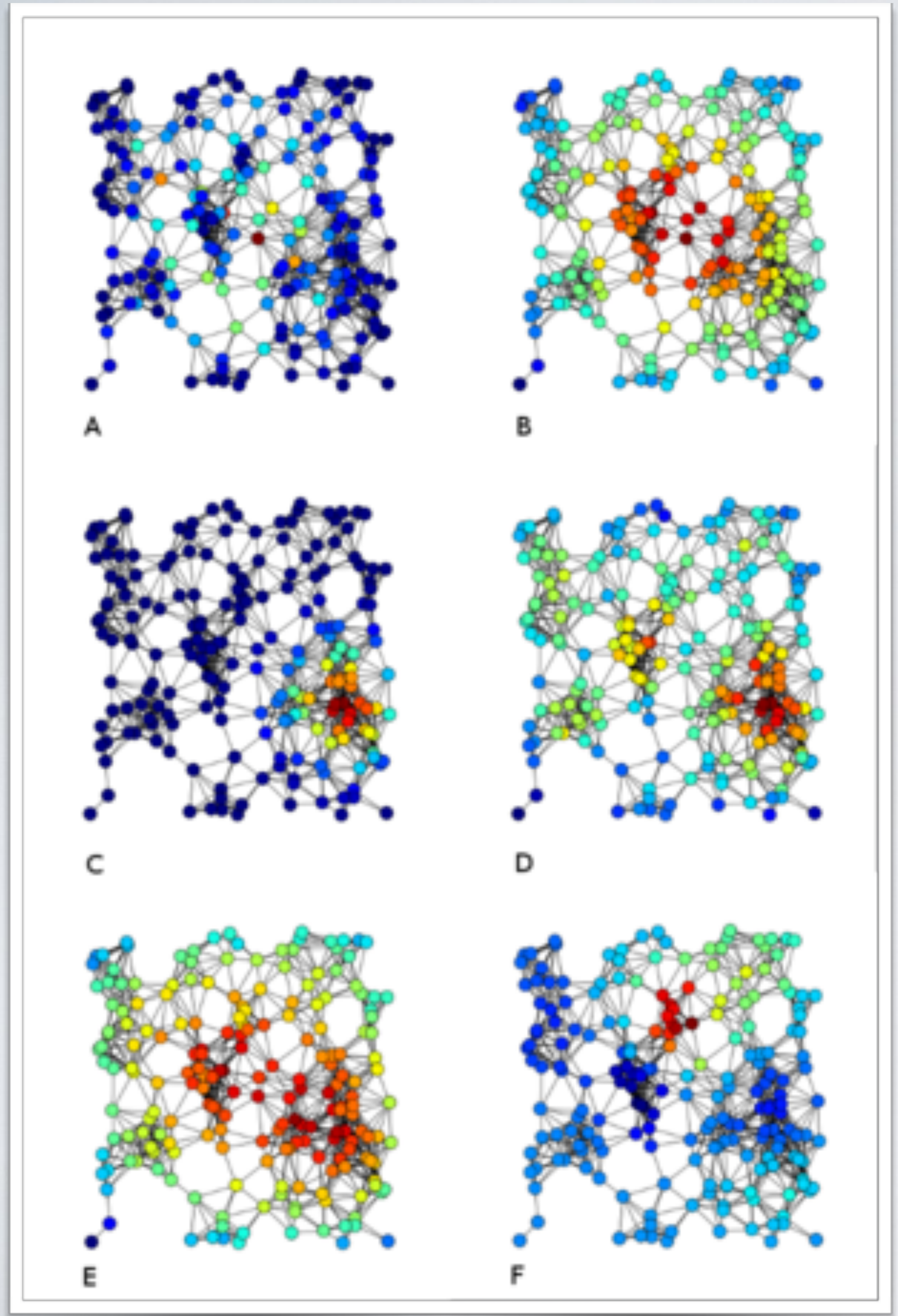
$$C(x) = \frac{1}{\sum_y d(y,x)}$$

Closeness

- **Harmonic centrality** related to closeness centrality

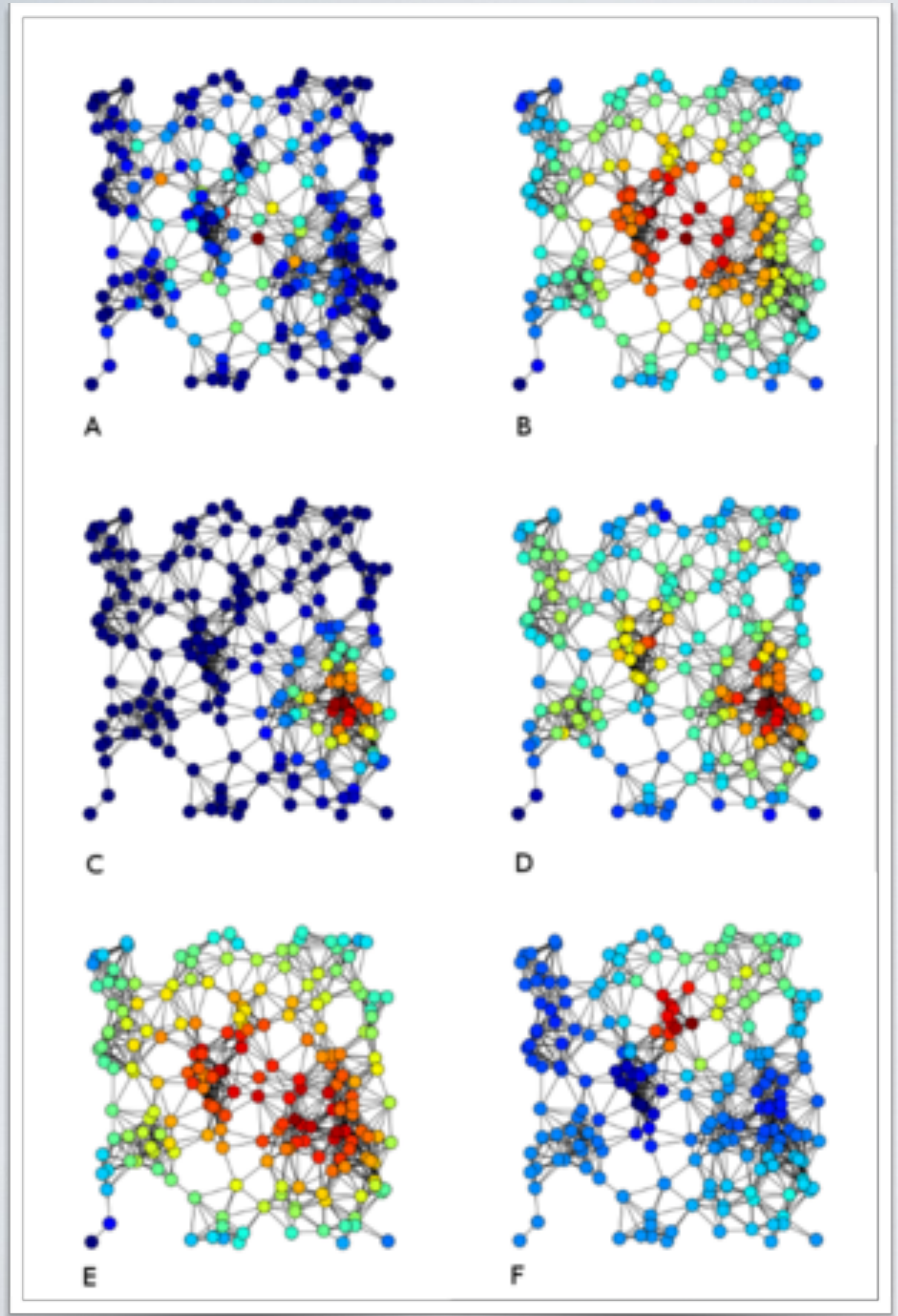
OTHERS

- Many other centralities have been proposed
- The problem is how to interpret them ?
- Can be used as supervised tool:
  - Compute many centralities on all nodes
  - Learn how to combine them to find chosen nodes
  - Discover new similar nodes
  - (roles in social networks, key elements in an infrastructure, …)

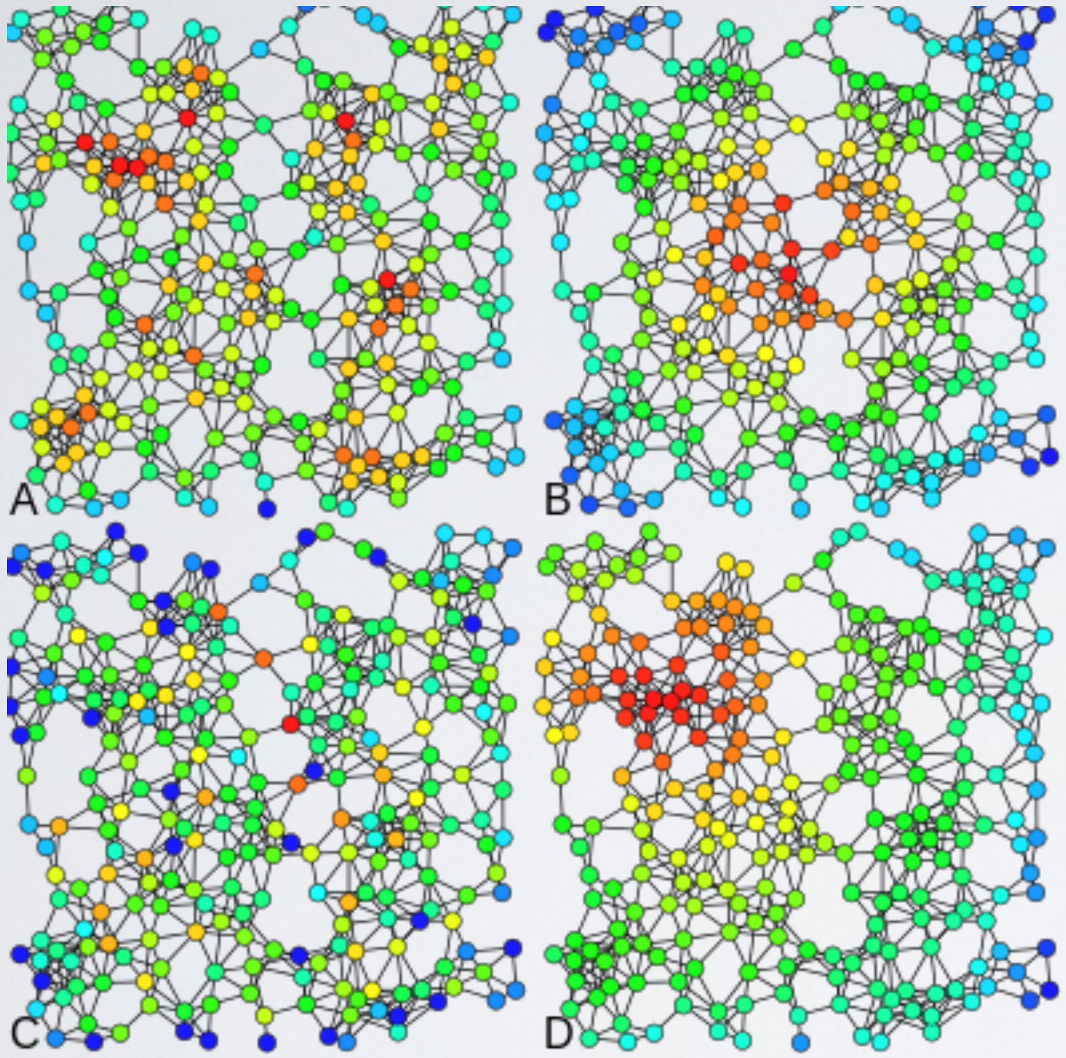


? Which is which

Harmonic
Closeness
Betweenness
Eigenvector
Katz
Degree



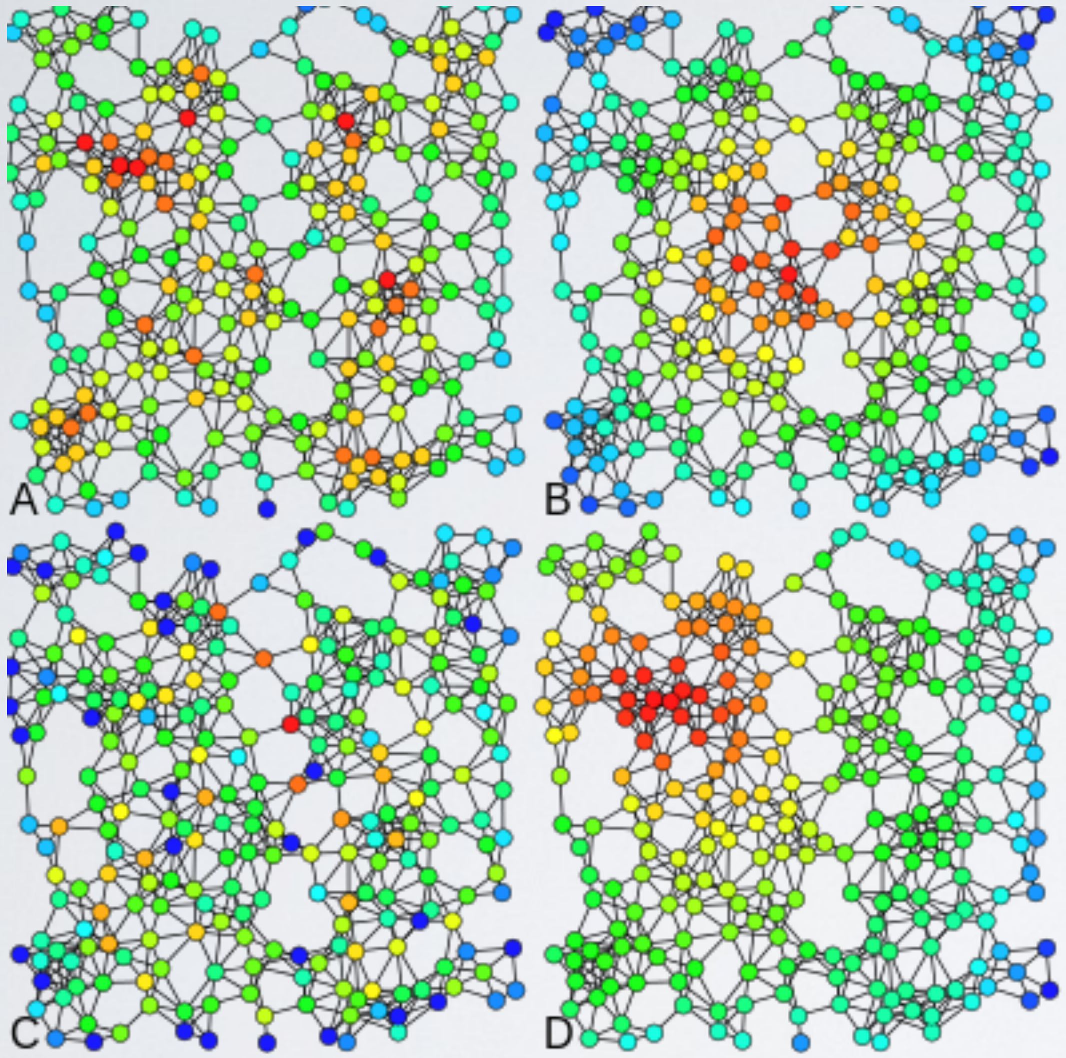
A: Betweenness
B:Closeness
C:Eigenvector
D:Degree
E:Harmonic
F: Katz



Try again :)

Degree
Betweenness
Closeness
Eigenvector

Try again :)



A: Degree
B:Closeness
C: Betweenness
D: Eigenvector

EDGES CENTRALITIES

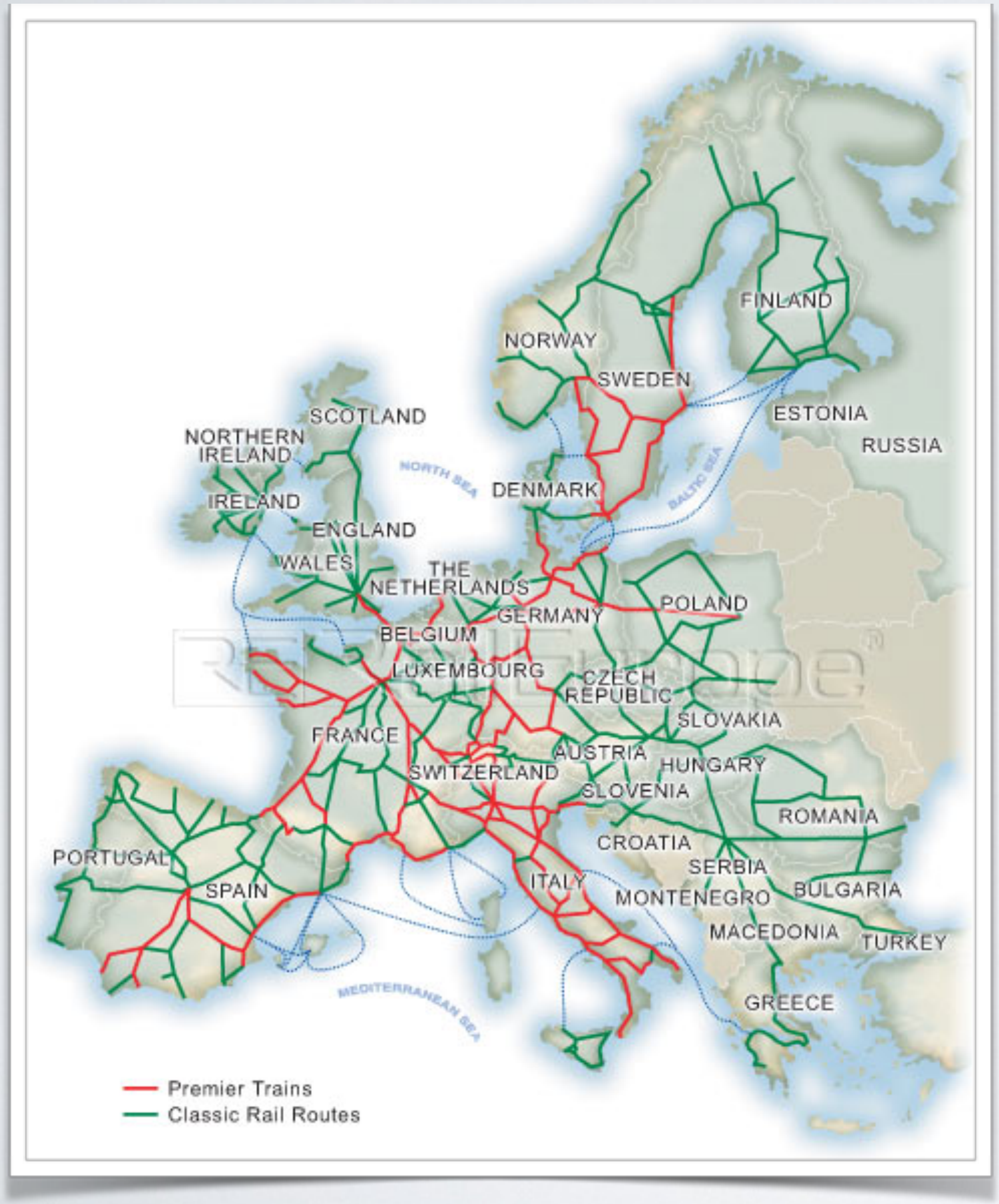
EDGES

- Most centralities can be computed for edges
- Methods based on flow are more natural for edges:
  - **Edge betweenness centrality**: how many shortest paths go through the **edge**

$$C_{B_e}(e) = \sum_{s \neq t \in V} \frac{\sigma_{st}(e)}{\sigma_{st}}$$

sigma: shortest paths

EDGES



Can you guess the edges of highest betweenness in the European rail network ?

K-PATH EDGE CENTRALITY

$$L^{\kappa}(e) = \sum_{s \in V} \frac{\sigma_s^{\kappa}(e)}{\sigma_s^{\kappa}}$$

s: source node

K-path: random walk of distance k

CURRENT-FLOW BETWEENNESS

Analogy with electrical circuit

How much voltage at the node if unit injected at random node and collected at other random node (average)

$$I_i^{(st)} = \frac{1}{2}\sum_j A_{ij}|V_i^{(st)} - V_j^{(st)}| = \frac{1}{2}\sum_j A_{ij}|T_{is} - T_{it} - T_{js} + T_{jt}|, \quad \text{for } i \neq s, t. \quad (9)$$

The current flowing through the ith vertex is given by a half of the sum of the absolute values of the currents flowing along the edges incident on that vertex

$$b_i = \frac{\sum_{s<t} I_i^{(st)}}{\frac{1}{2}n(n-1)}.$$

Average current flow for all pairs

CURRENT-FLOW BETWEENNESS

Also called Random walk betweenness

Average probability to go through the edge
in a random walk from
U to V for all pairs of nodes (U,V)

COMMUNICABILITY BETWEENNESS CENTRALITY

$$G_{pq} = \frac{1}{s!} P_{pq}^{(s)} + \sum_{k>s} \frac{1}{k!} W_{pq}^{(k)}$$

$$\omega_r = \frac{1}{C} \sum_p \sum_q \frac{G_{prq}}{G_{pq}}, \qquad p \neq q, p \neq r, q \neq r$$

COMMUNICABILITY BETWEENNESS CENTRALITY

$$G_{pq} = \frac{1}{s!}\boxed{P^{(s)}_{pq}} + \sum_{k>s} \frac{1}{k!} W^{(k)}_{pq}$$

Number of shortest paths of length< **s**

$$\omega_r = \frac{1}{C}\sum_p \sum_q \frac{G_{prq}}{G_{pq}}, \qquad p \neq q, p \neq r, q \neq r$$

COMMUNICABILITY BETWEENNESS CENTRALITY

$$G_{pq} = \frac{1}{s!} P_{pq}^{(s)} + \sum_{k>s} \frac{1}{k!} \boxed{W_{pq}^{(k)}}$$

Number of shortest paths of length> **s**

$$\omega_r = \frac{1}{C} \sum_p \sum_q \frac{G_{prq}}{G_{pq}}, \qquad p \neq q, p \neq r, q \neq r$$

COMMUNICABILITY BETWEENNESS CENTRALITY

$$G_{pq} = \frac{1}{s!}P_{pq}^{(s)} + \sum_{k>s}\frac{1}{k!}W_{pq}^{(k)}$$

$$\omega_r = \frac{1}{C}\sum_p\sum_q \boxed{\frac{G_{prq}}{G_{pq}}}, \qquad p \neq q, p \neq r, q \neq r$$

Score for paths going though r / scores for all paths

EDGE CENTRALITIES

- And of course, many more

COMPLEXIFYING COMPLEX NETWORKS

WEIGHTS, DIRECTIONS

- Until now, I presented mostly undirected, unweighted networks
- Most of what we have seen generalizes naturally to:
  - Weighted (value on edges)
  - Directed (edges in a single direction)

WEIGHTS

- **Degree** on a weighted network is called **strength**
  - Sum of weights of edges
- Random walks are **biased** according to weights
- Sone notions are harder: Clustering Coefficient? Graph Distance?

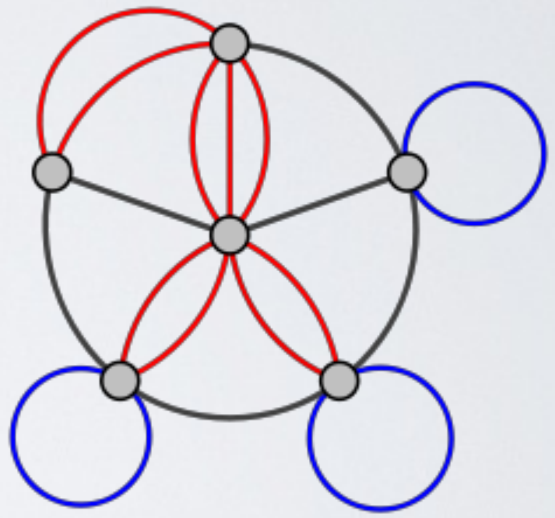
DIRECTIONS

- **Degree** is split between **in-degree** and **out-degree**
- Random walks naturally follow directions
- Shortest paths naturally defined
- Modularity is problematic, clustering coefficient has several definitions…

MULTI-GRAPHS

MULTIGRAPH

- Multi-graph: several edges allowed between same nodes
- Often used in conjunction with labels:
  - One edge for friendship,
  - one edge for family,
  - one edge for co-worker…



- Without labels, can be simplified as a weighted graph

MULTI-PARTITE GRAPHS

MULTI-PARTITE GRAPHS

- Bi-partite: there exists 2 kinds of nodes, and links can be only between nodes of different types
  - Multi-partite: similar but with more than 2 types. Much less common
- Not strictly different from normal graphs: if you don't know the two categories of nodes, it looks like any network
- The problem is that some definitions of normal graphs become meaningless
  - =>Clustering coefficient

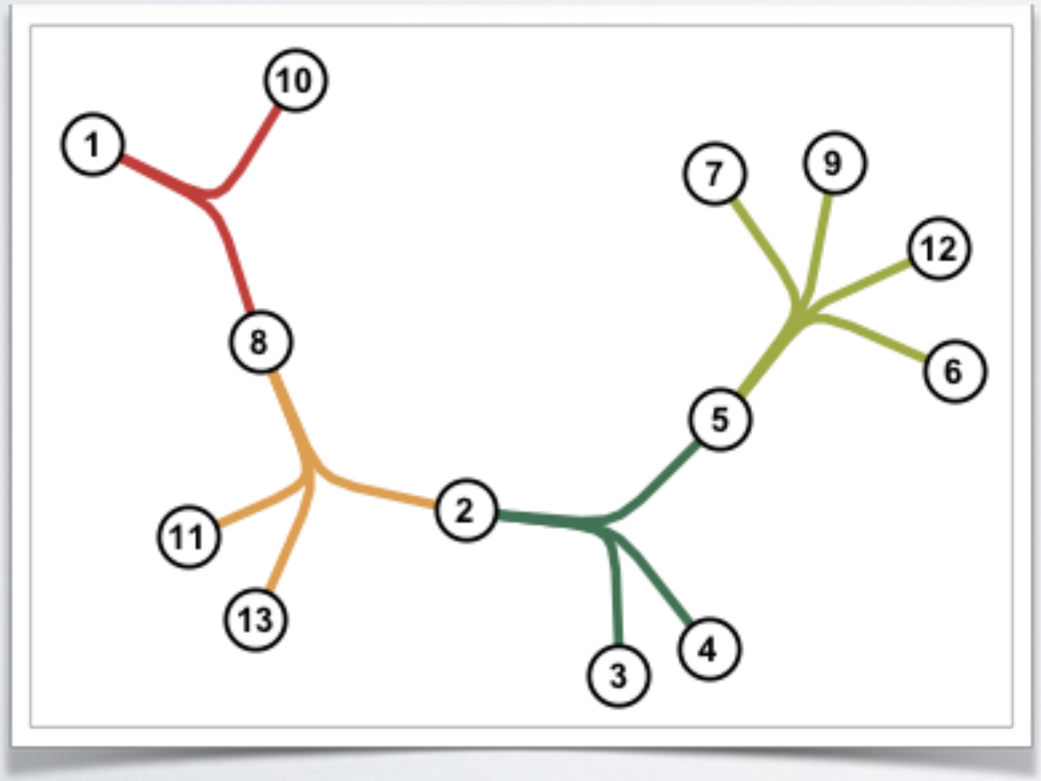
MULTI-PARTITE GRAPHS

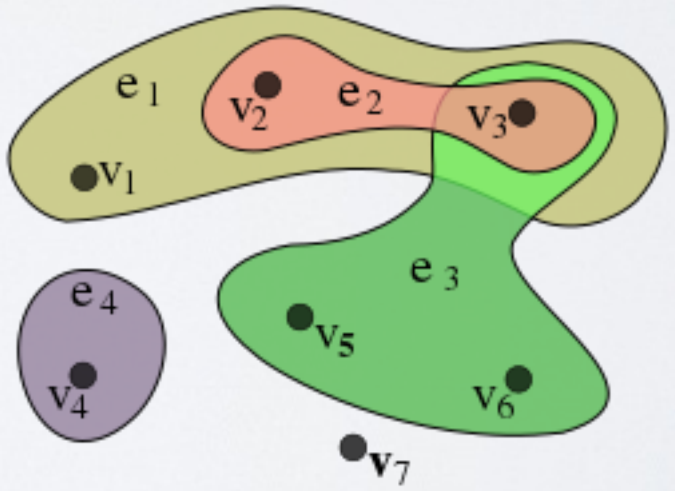
- Bi-partite networks are quite commonly use
  - Actors - Films
  - Clients - Products
  - Researchers - conferences/institutions
  - …
- Normal methods work but sometimes give unintuitive results:
  - Specific variants have been proposed

HYPERGRAPHS

HYPERGRAPHS

- "Generalisation" of graph
- An edge is not limited to 2 extremities





MULTILAYER NETWORKS

MULTILAYER NETWORKS

- Multiplex network
- Multislice network
- Multitype network
- Heterogenous information network

[Kivela 2014]

MULTILAYER NETWORKS

- Can be used to represent:
  - Several types of relations between the same nodes
    - Bus transportation network
    - Bicycle transportation network
    - Car transportation network
    - …

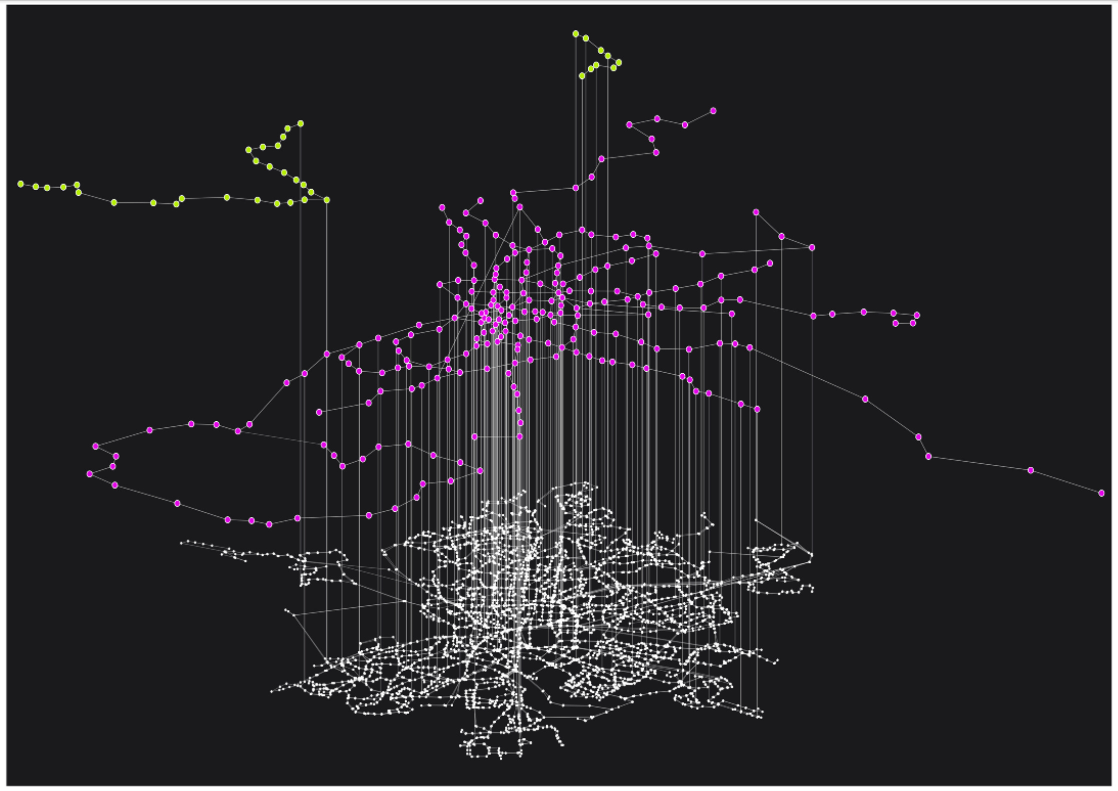
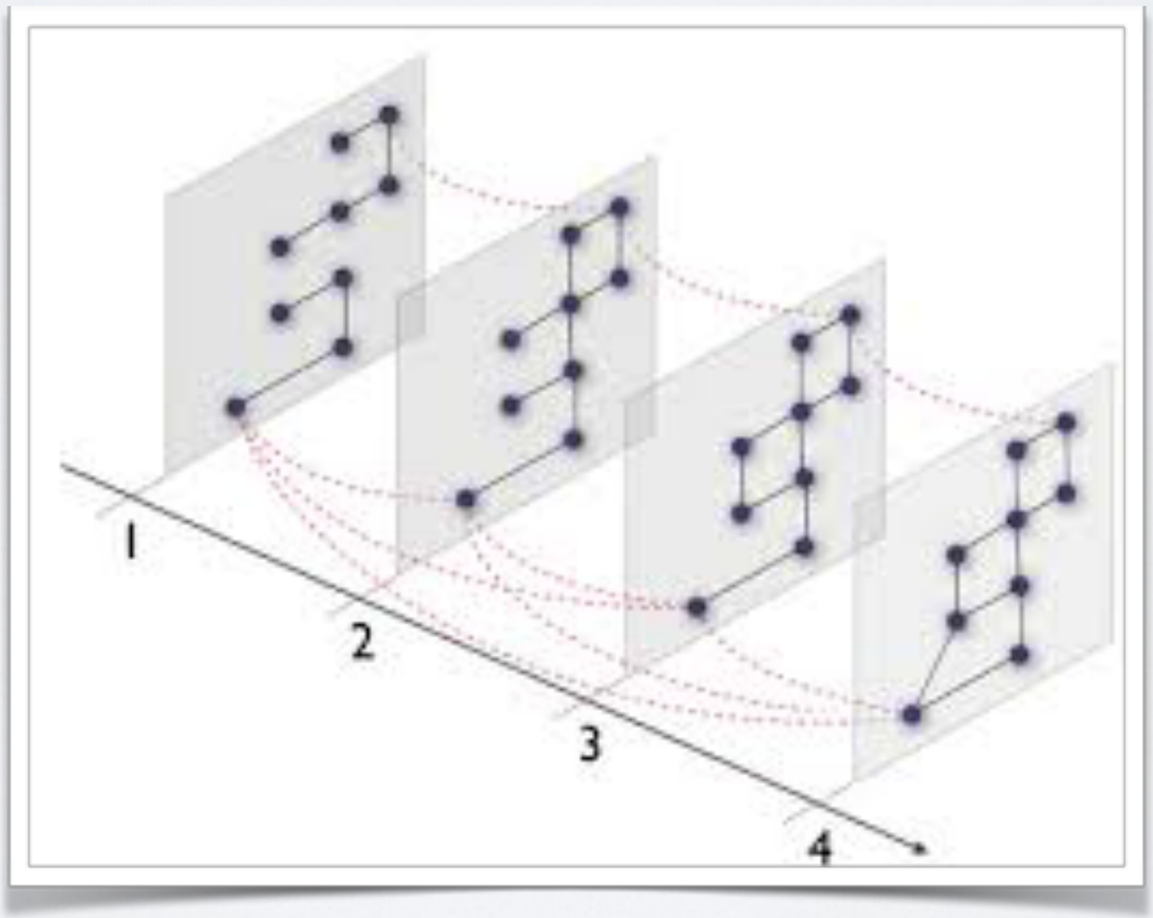


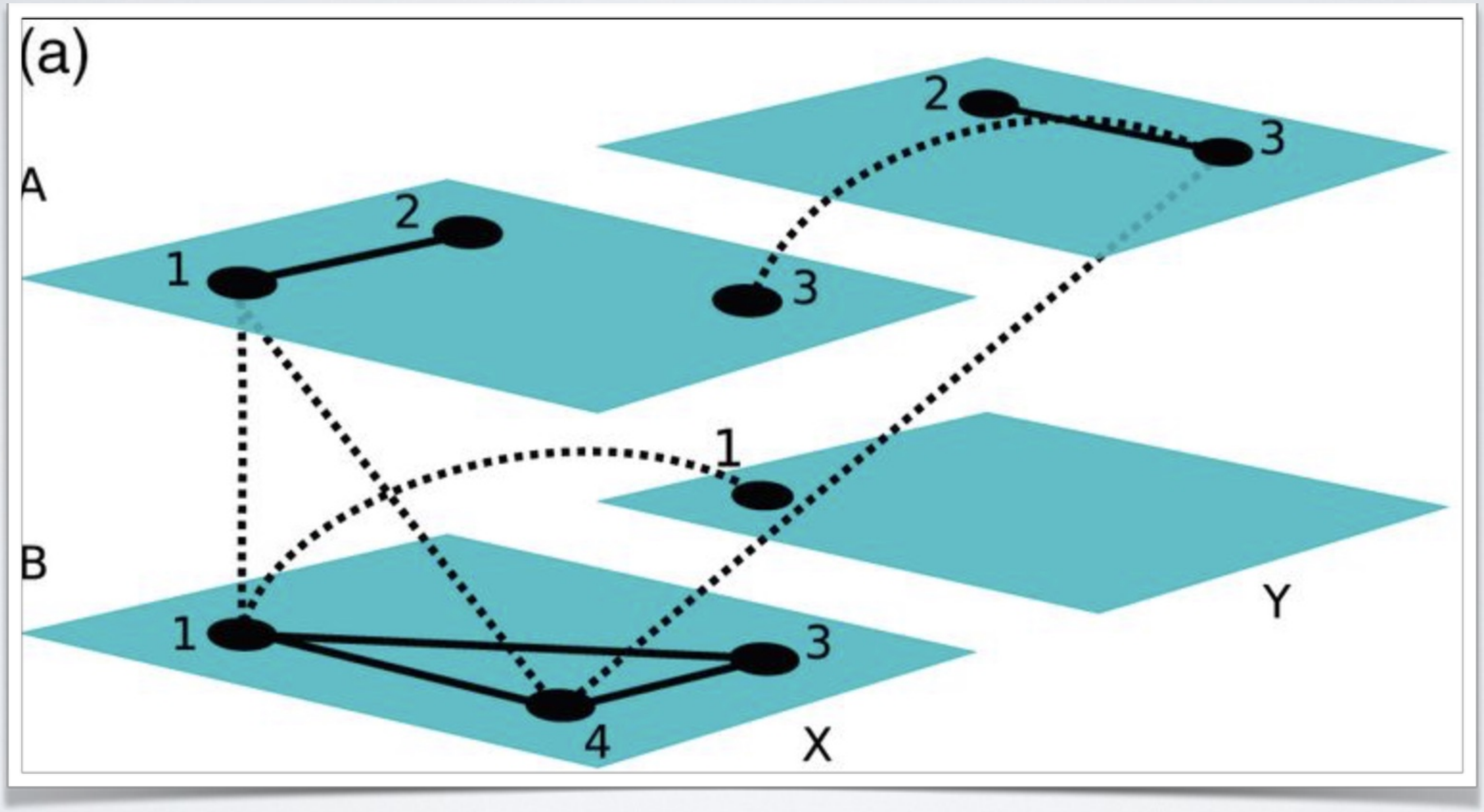
Figure 2. Superlayer representation of the Madrid transportation system. The figure represents the three transportation modes considered: tram (yellow nodes, upper layer), metro (purple nodes, mid layer) and buses (white nodes, bottom layer). See Table1 for statistics of these layers.

MULTILAYER NETWORKS

- Can be used to represent:
  - Several snapshots of the same network



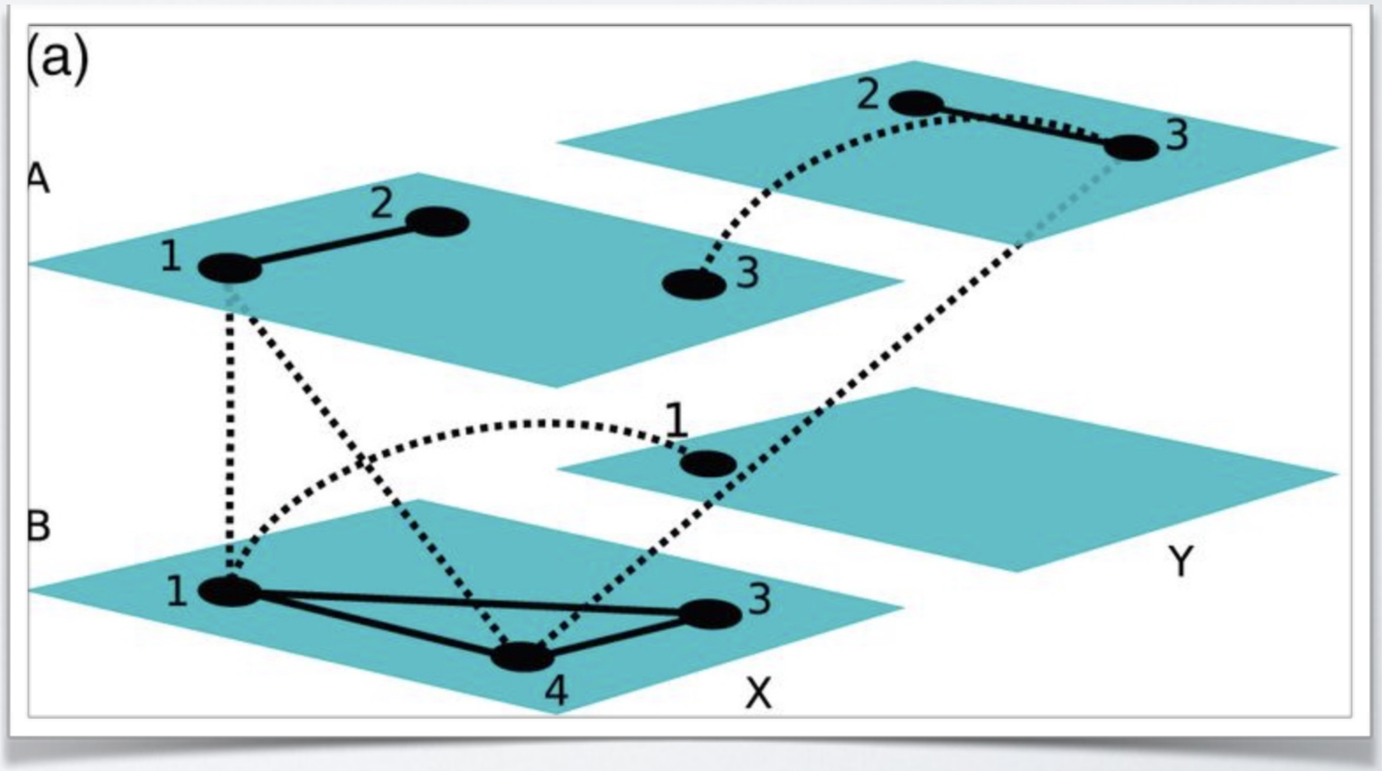
MULTILAYER NETWORKS



Both/Other

MULTILAYER NETWORKS

- Relations can be:
  - Only between **same** nodes in different layers
    - Public transport interconnection
  - Between different nodes in different layers
    - Information transfert form person A on Facebook to person B on Instagram.

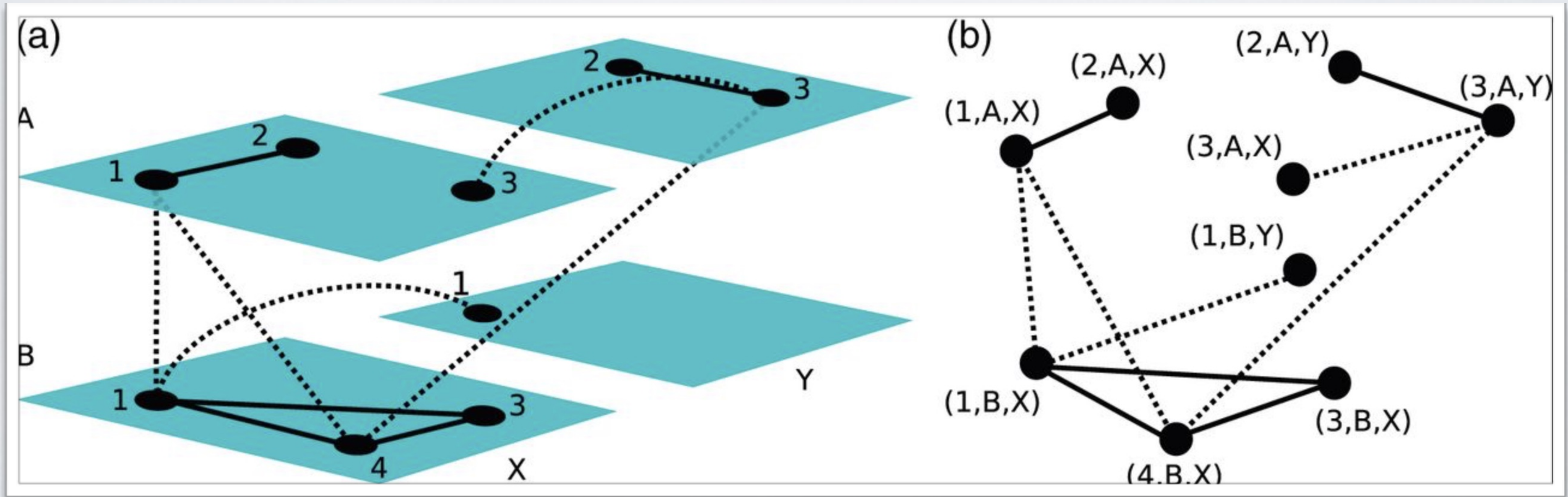


MULTILAYER NETWORKS

- All usual definitions on static networks can be extended to multilayer networks
  - Degree, clustering coefficient, community detection…
- The problem is that there are many ways to do it, and it depends on what your layers represent
  - Degree of a person on a multilayer network of facebook, Twitter, Linked-in?
- If you used a multilayer network, it is because it was not well summarized by a single network…
  - Same definition for multilayer dynamic and multilayer different types?

MULTILAYER NETWORKS

A simple idea: multilayers networks can be transformed into normal networks



HIGHER ORDER NETWORKS

HIGHER ORDER NETWORKS

- Another relatively recent and very active field of research
- Many networks are built using logs of *sequence of items* encountered by *actors*
  - People travelling in public transport go through stations
  - Consumer buy products on amazon one after the other
- Normal network: we split sequences in pairs
  - Higher order: conserve the memory of previous items

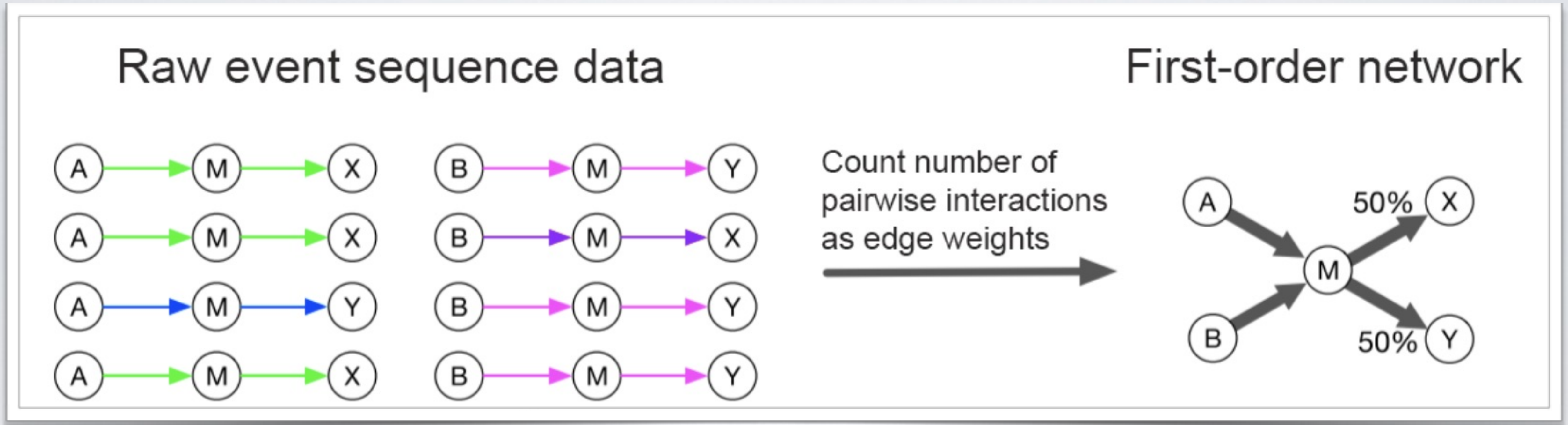
HIGHER ORDER NETWORKS

- Typical example: air traffic.
- Many cities does not have direct trips
  - e.g.: Lyon->New York
  - Flight goes through stopover: Lyon->Paris->New York
- If we want to create a weighted network of trips:
  - We count the number of trips Lyon->Paris, and Paris->New York
  - We forget the information of previous/Next step

HIGHER ORDER NETWORKS

- But information of previous steps can be useful!
- From Paris,
  - 10% passengers go to New York
  - 10% passengers go to Rome
- You know that a passenger comes from Lyon
  - 10% probability to go to New York
  - 10% probability to go to Rome
  - =>Wrong!
  - If I come from lyon, much more likely to go to New York than Rome!

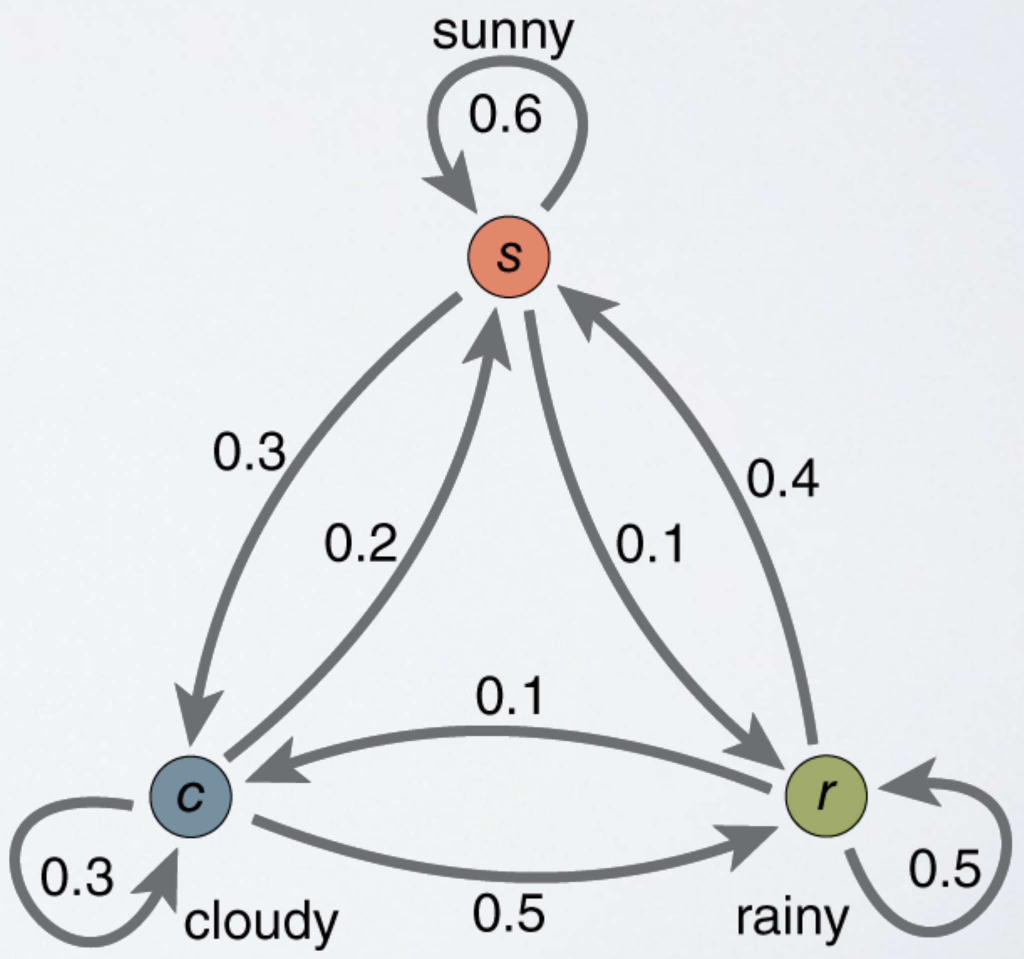
HIGHER ORDER NETWORKS



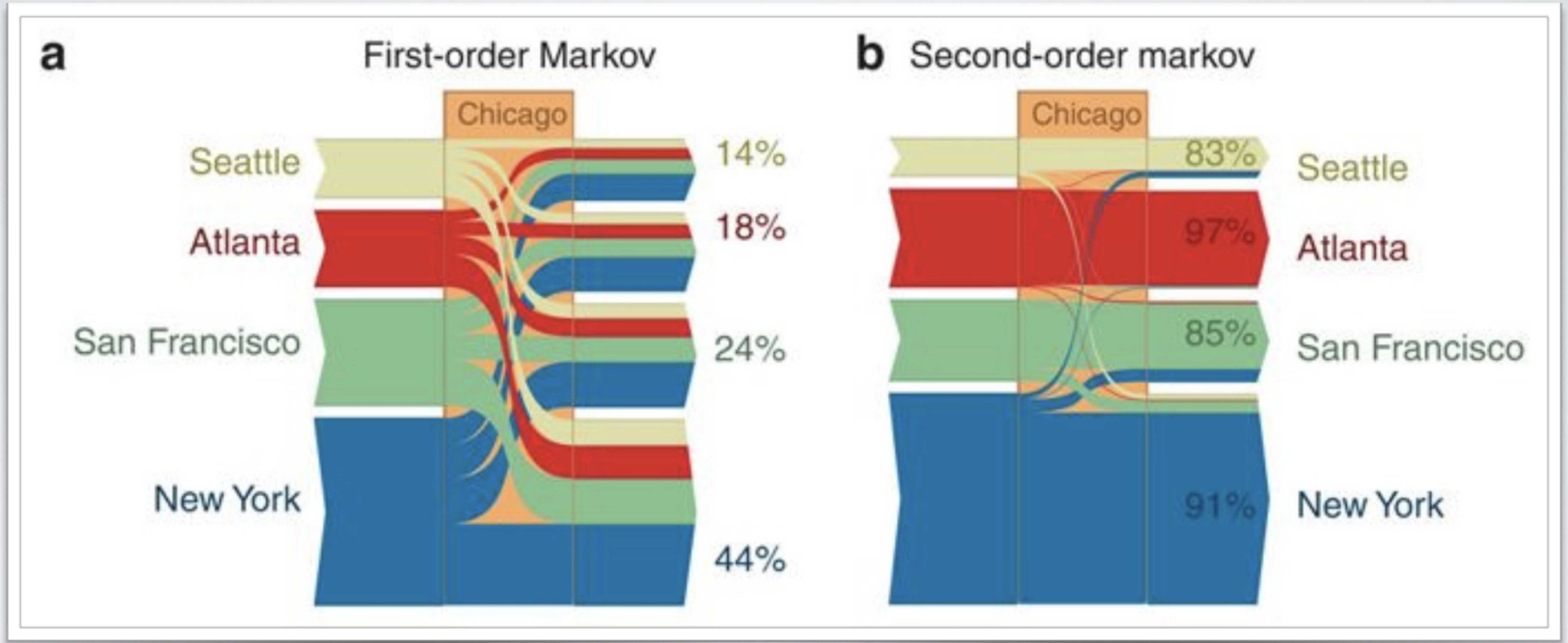
[Xu 2016]

HIGHER ORDER NETWORKS

- The big word: **Non-markovian process**
- Markovian process:
  - A random process in which the future is independent from the past
  - Describes a process: Markov chains



HIGHER ORDER NETWORKS



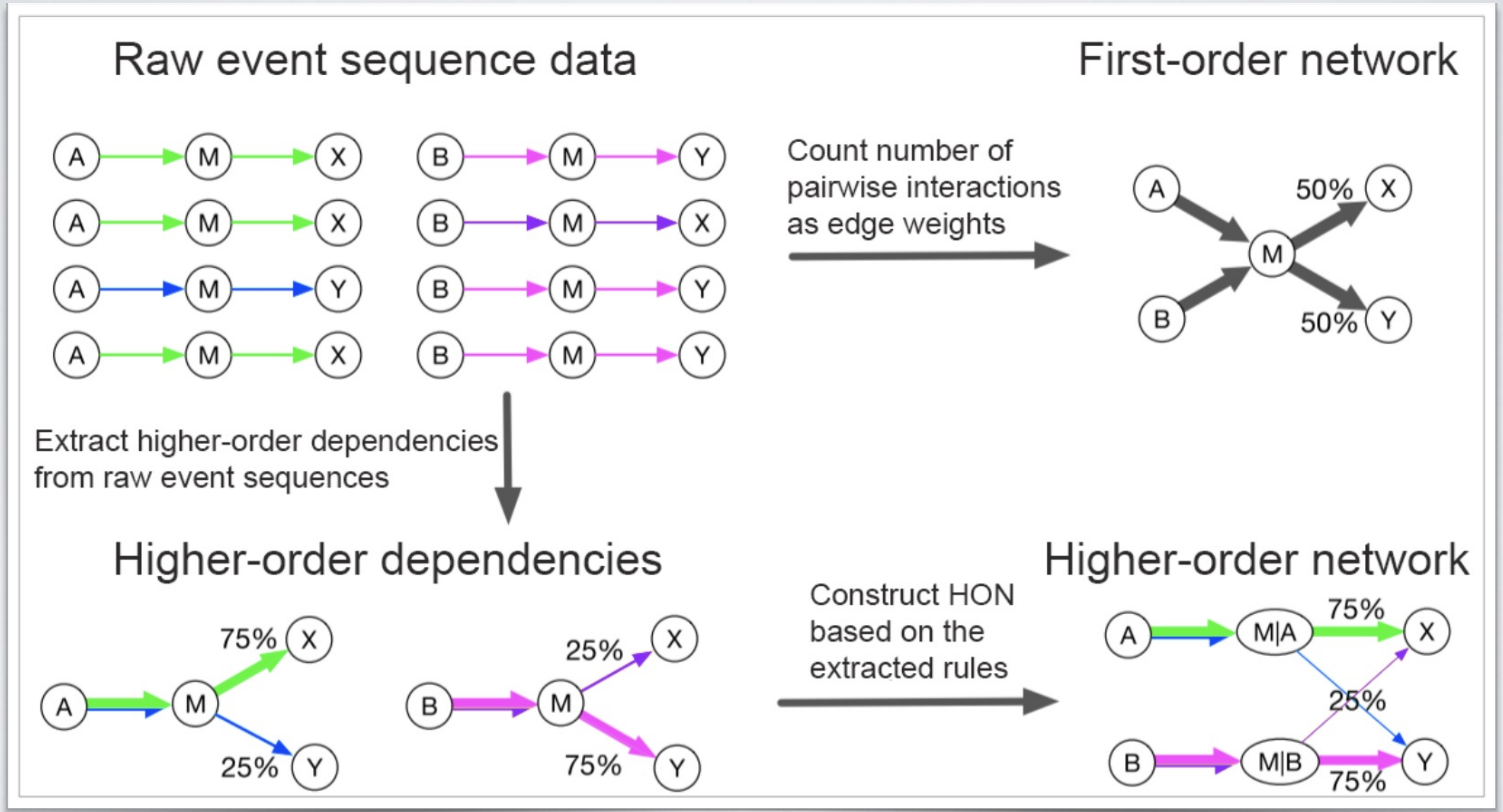
Round trips

[Rosvall 2014]

HIGHER ORDER NETWORKS

- How to integrate non-markovian processes in networks?
- We create new nodes
  - Do not correspond to an element (a city…)
  - Correspond to an element AND an origin
- A single element of memory: second-order network
- 2? Third-order network
- etc.

HIGHER ORDER NETWORKS


Raw event sequence data
First-order network
Count number of pairwise interactions as edge weights
Extract higher-order dependencies from raw event sequences
Higher-order dependencies
Higher-order network
Construct HON based on the extracted rules
A
B
M
X
Y
M|A
M|B
50%
75%
25%

HIGHER ORDER NETWORKS

- Weakness: complexity
- You multiply the number of nodes by the number of possible arrival source
- => Can be necessary to ignore rare cases

MANIPULATING GRAPHS: GEPHI, NETWORKX

MANIPULATING GRAPHS

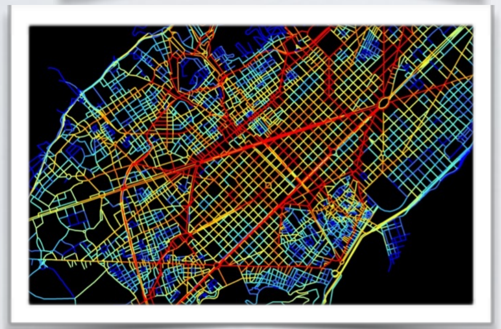
- Visualization + simple computations
  - Gephi
  - Tulip
  - …
- Libraries
  - Networkx (simple, most complete)
  - SNAP (fast)
  - igraph, graph-tools, Apache GraphX, … (useful for specific tasks)

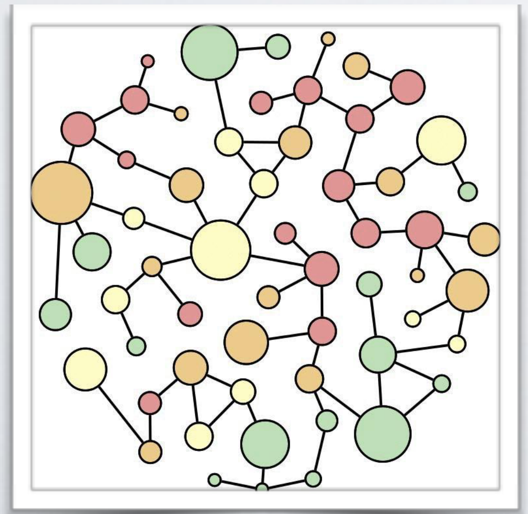
MANIPULATING GRAPHS

- Demo

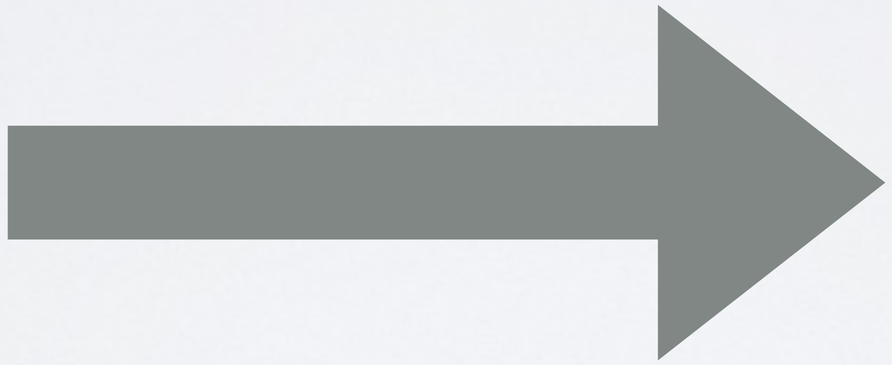
WHAT IS GRAPH EMBEDDING ?

GRAPHS / NETWORKS





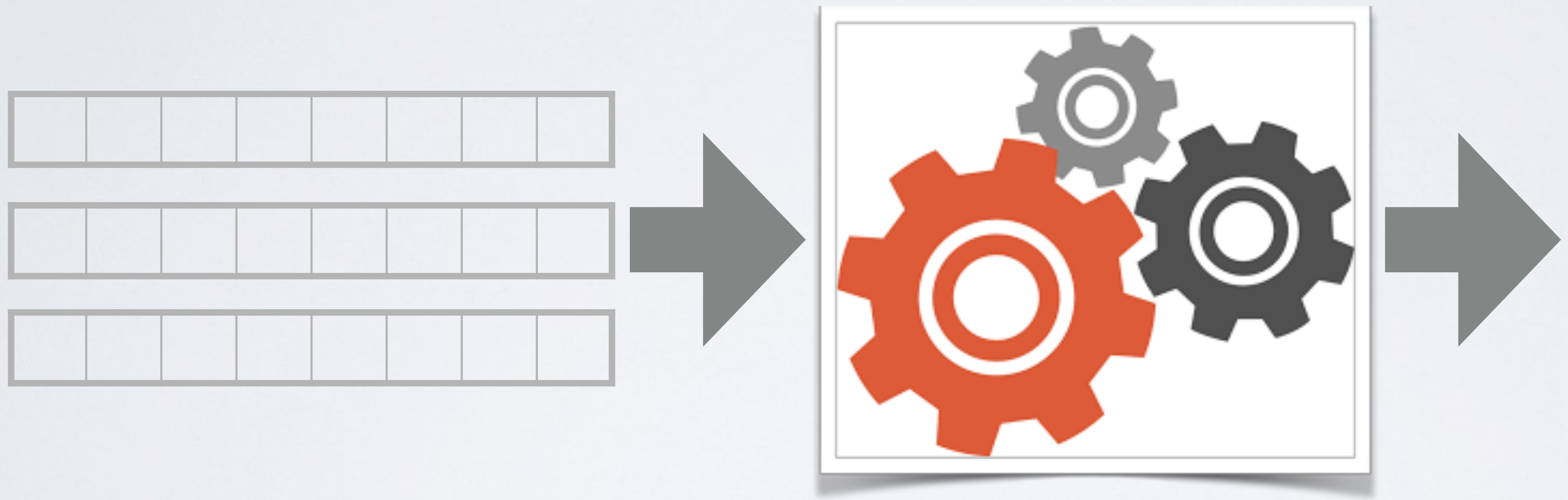
Ad Hoc
Network Algorithms



Link prediction
Community detection
Graph reconstruction
Node classification
…

MACHINE LEARNING

Features as vectors

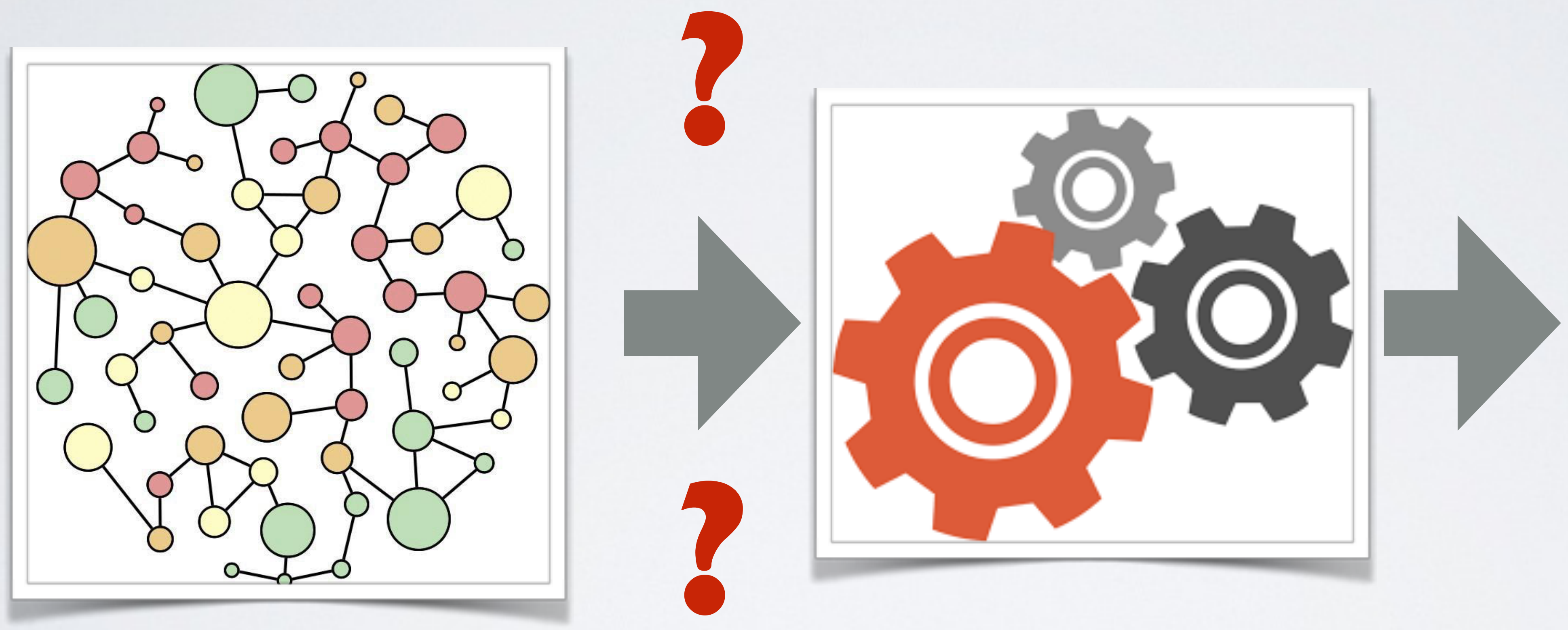


Prediction

Classification

Clustering

…



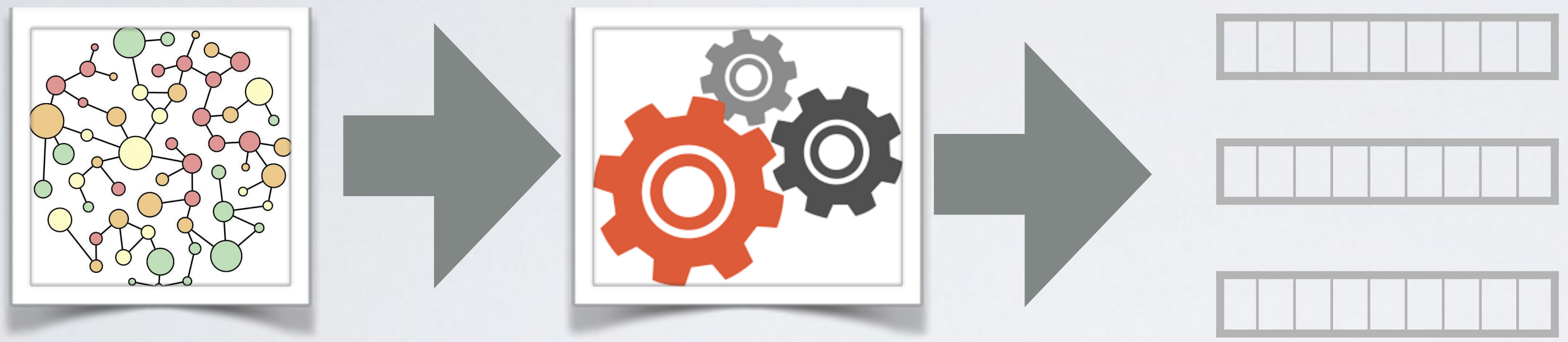
Prediction

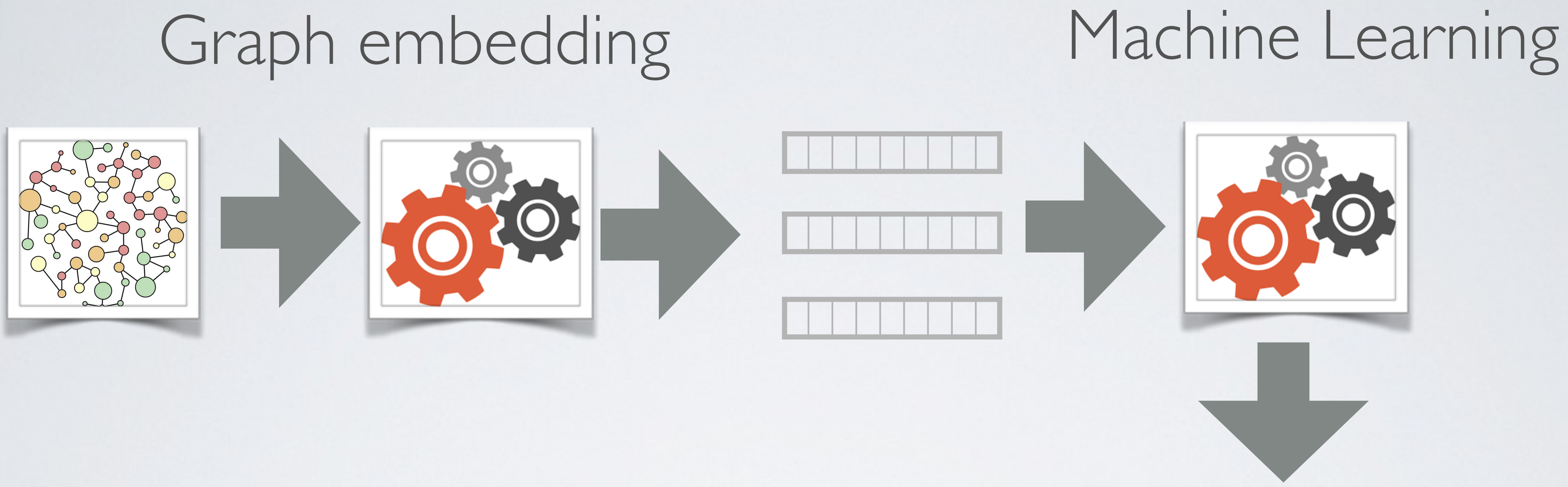
Classification

Clustering

…

Graph embedding

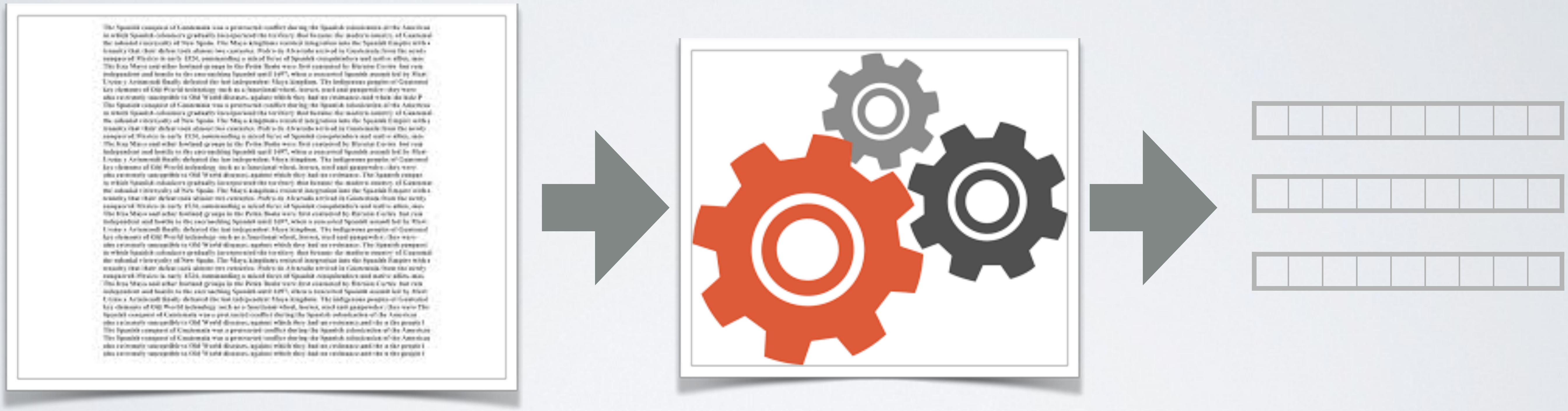




Link prediction Graph reconstruction	=	Prediction
Community detection	=	Clustering
Node classification	=	Classification
…		…

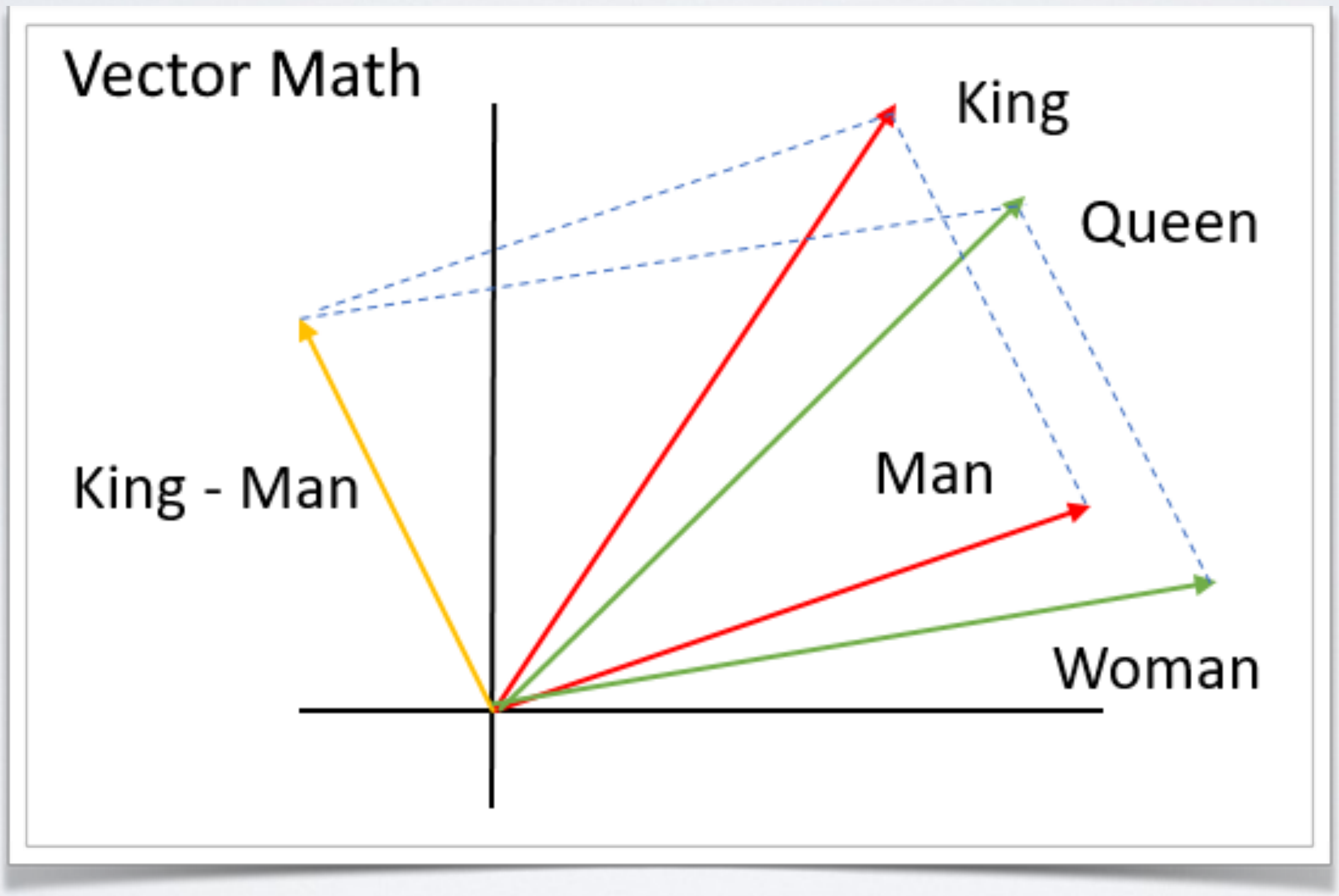
WHY DOES GRAPH EMBEDDING MATTERS ?

WORD EMBEDDING



WORD EMBEDDING

Word2vec, Skipgram, …



GENERIC “SKIPGRAM”

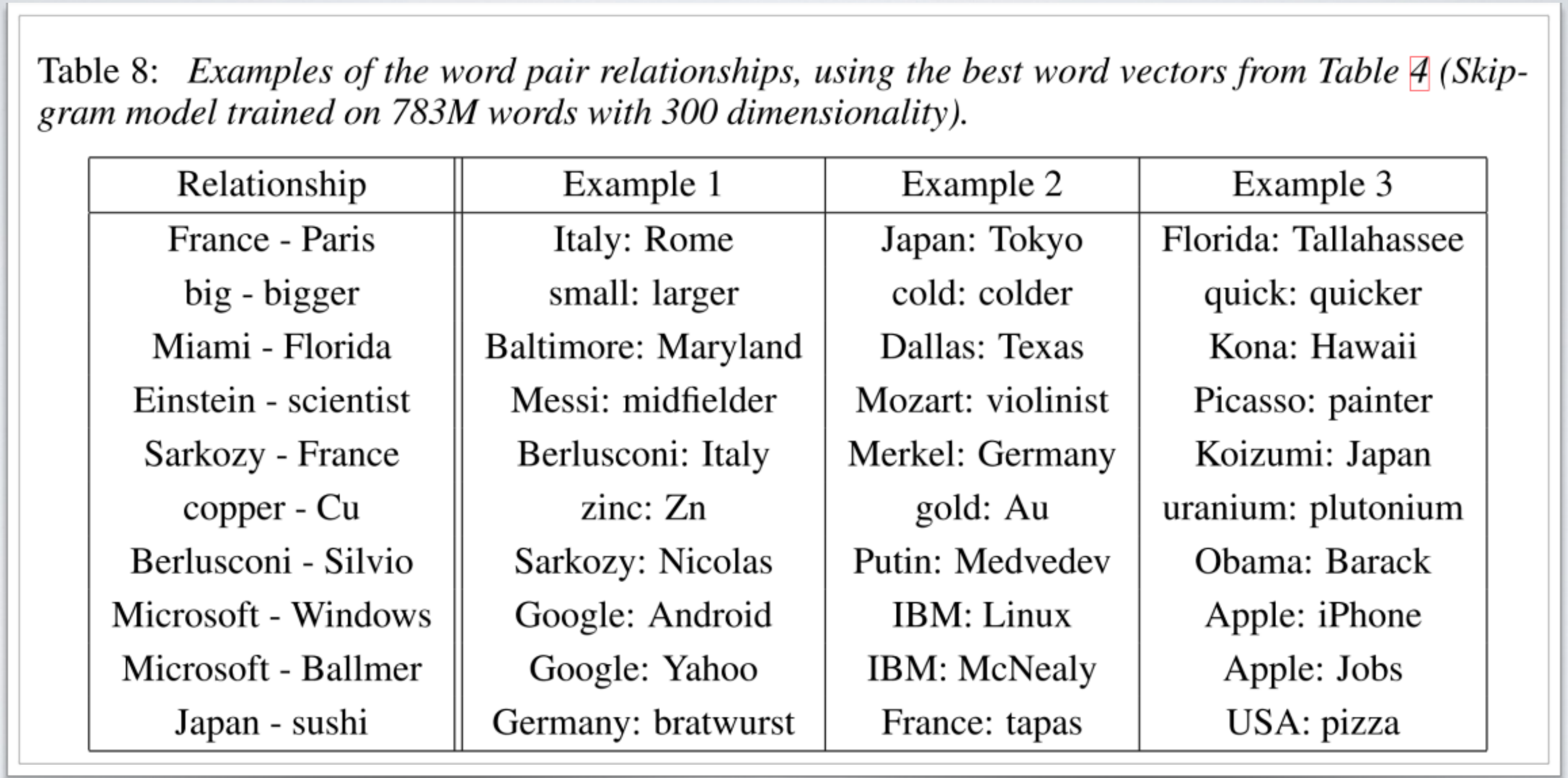


Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (Skip-gram model trained on 783M words with 300 dimensionality).*

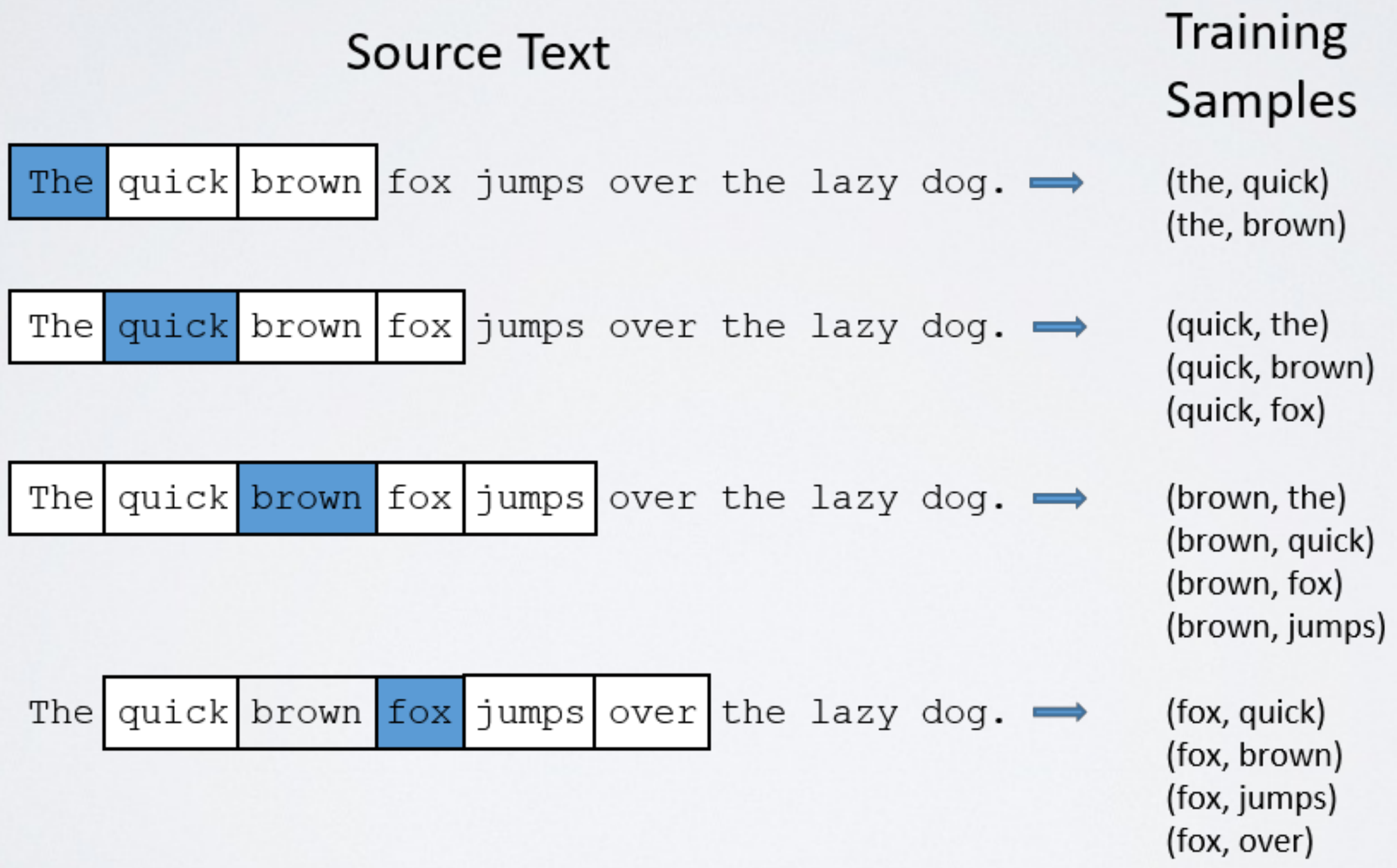
Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

[https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/]

HOW DOES IT WORKS ?

SKIPGRAM

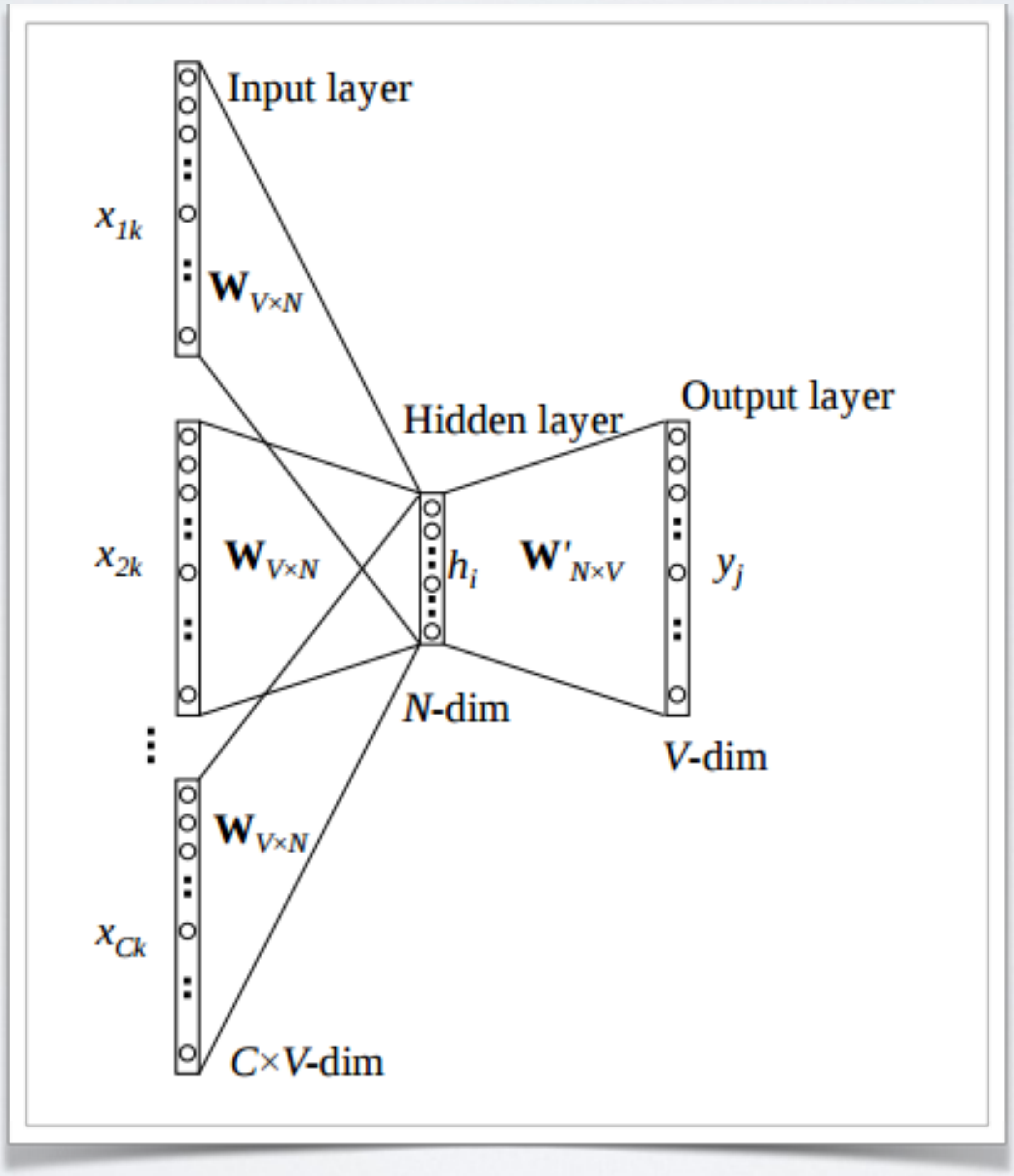
Word embedding
Natural language => vectors



[http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/]

INTUITIVE/NAIVE IDEA

- Recent methods based on "neural networks"



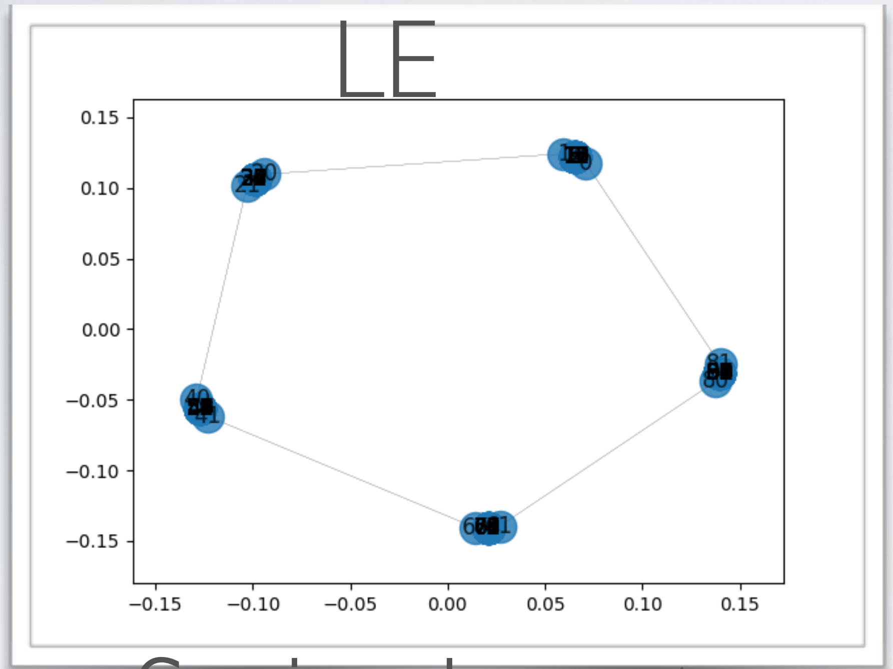
DEEPWALK

- Skipgram for graphs:
  - 1)Generate "sentences" using random walks
  - 2)Apply Skipgram
- Parameters: dimensions *d*, RW length *k*

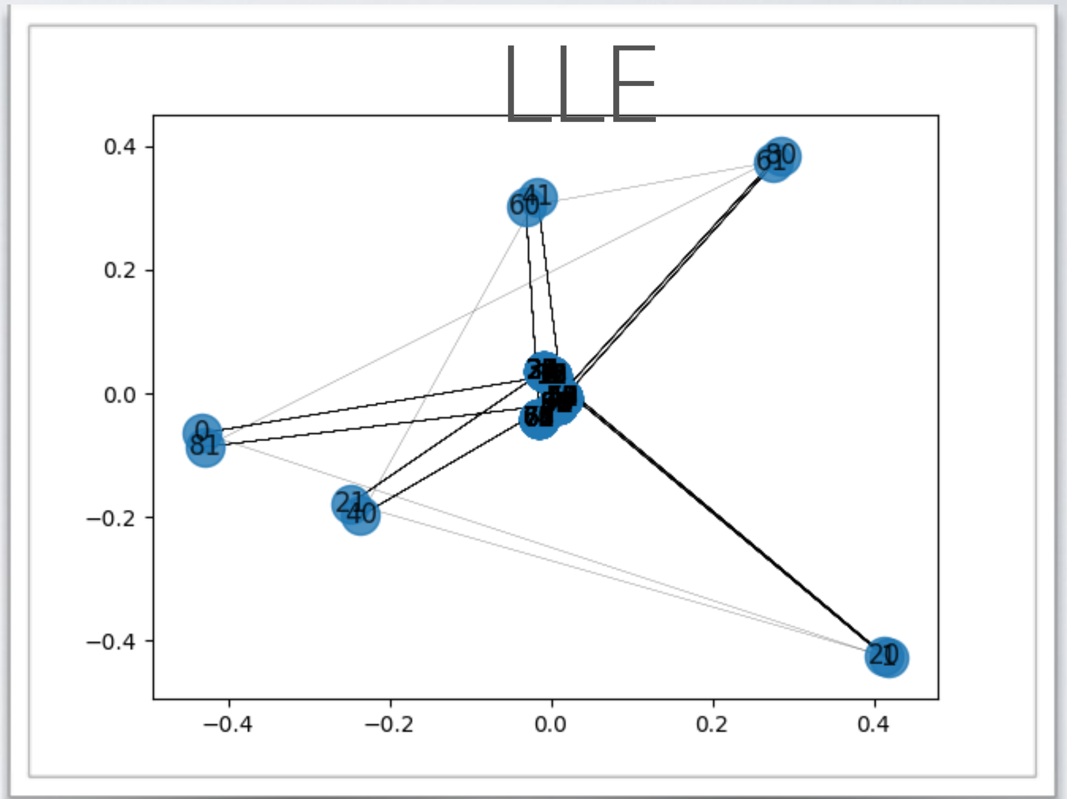
CLIQUE RING

5 cliques or size 20 with 1 edge between them

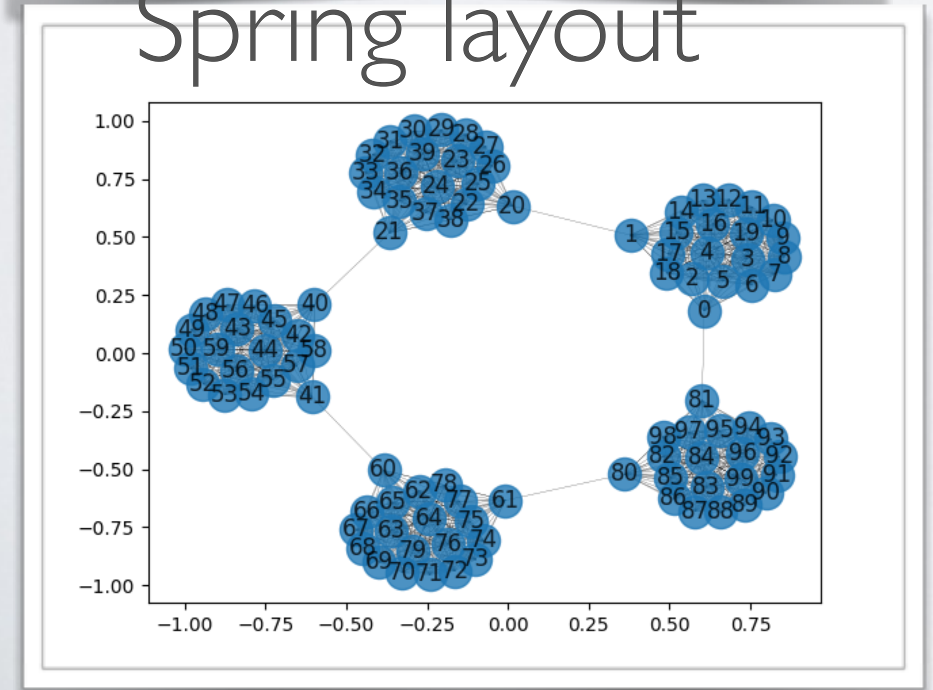
LE



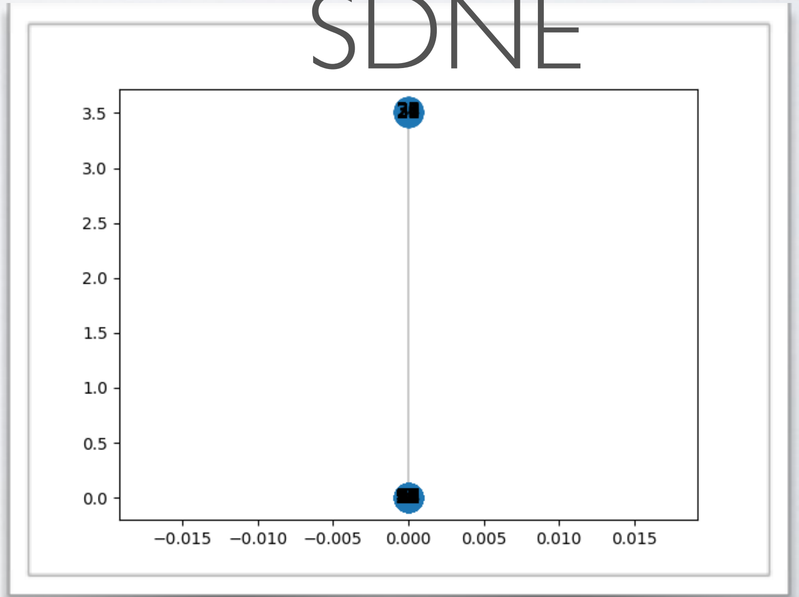
LLE



Spring layout

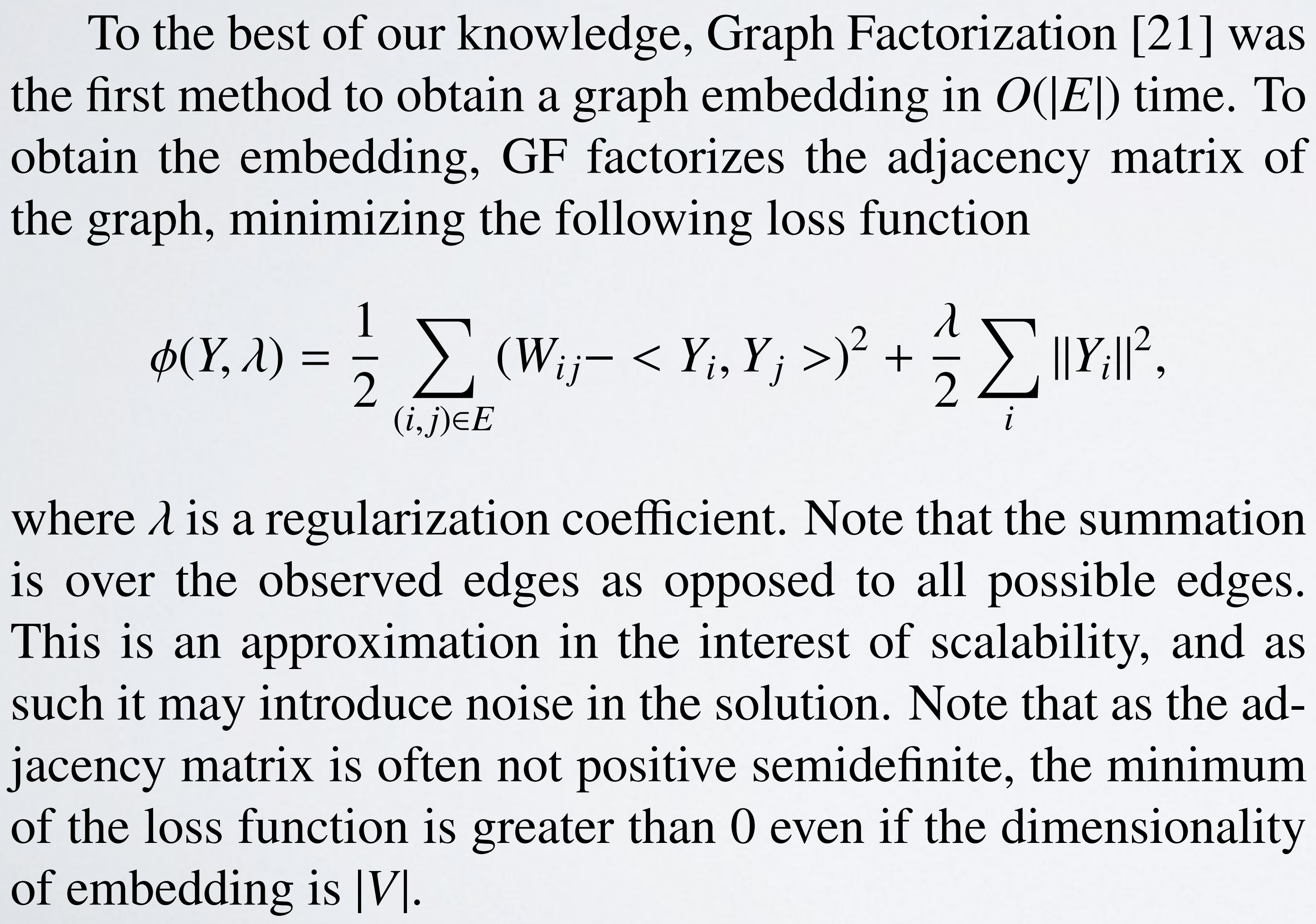


SDNE



GRAPH FACTORIZATION

- (published: 2013)



To the best of our knowledge, Graph Factorization [21] was the first method to obtain a graph embedding in $O(|E|)$ time. To obtain the embedding, GF factorizes the adjacency matrix of the graph, minimizing the following loss function

$$\phi(Y,\lambda) = \frac{1}{2}\sum_{(i,j)\in E}(W_{ij} - <Y_i, Y_j>)^2 + \frac{\lambda}{2}\sum_i \|Y_i\|^2,$$

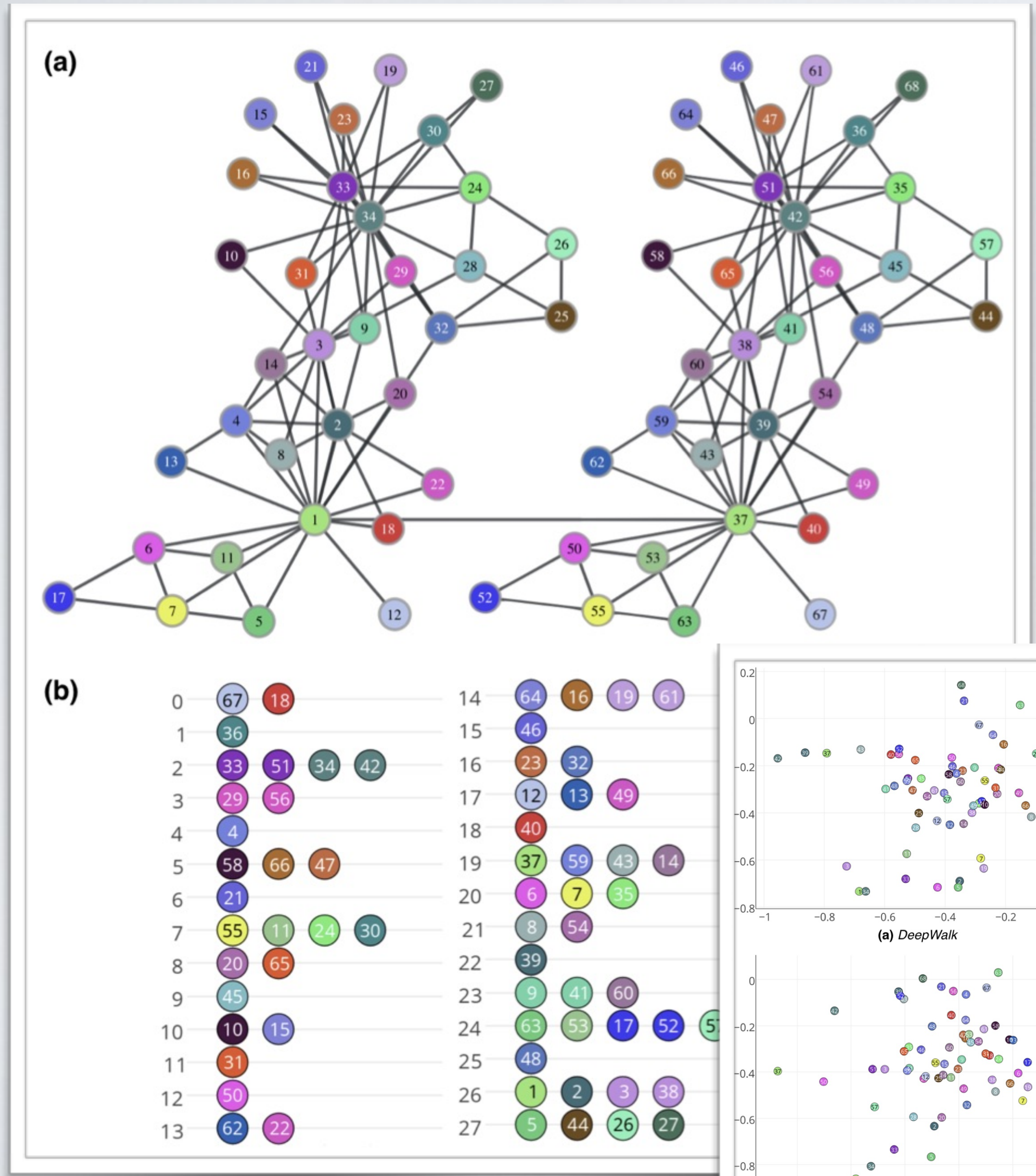
where λ is a regularization coefficient. Note that the summation is over the observed edges as opposed to all possible edges. This is an approximation in the interest of scalability, and as such it may introduce noise in the solution. Note that as the adjacency matrix is often not positive semidefinite, the minimum of the loss function is greater than 0 even if the dimensionality of embedding is $|V|$.

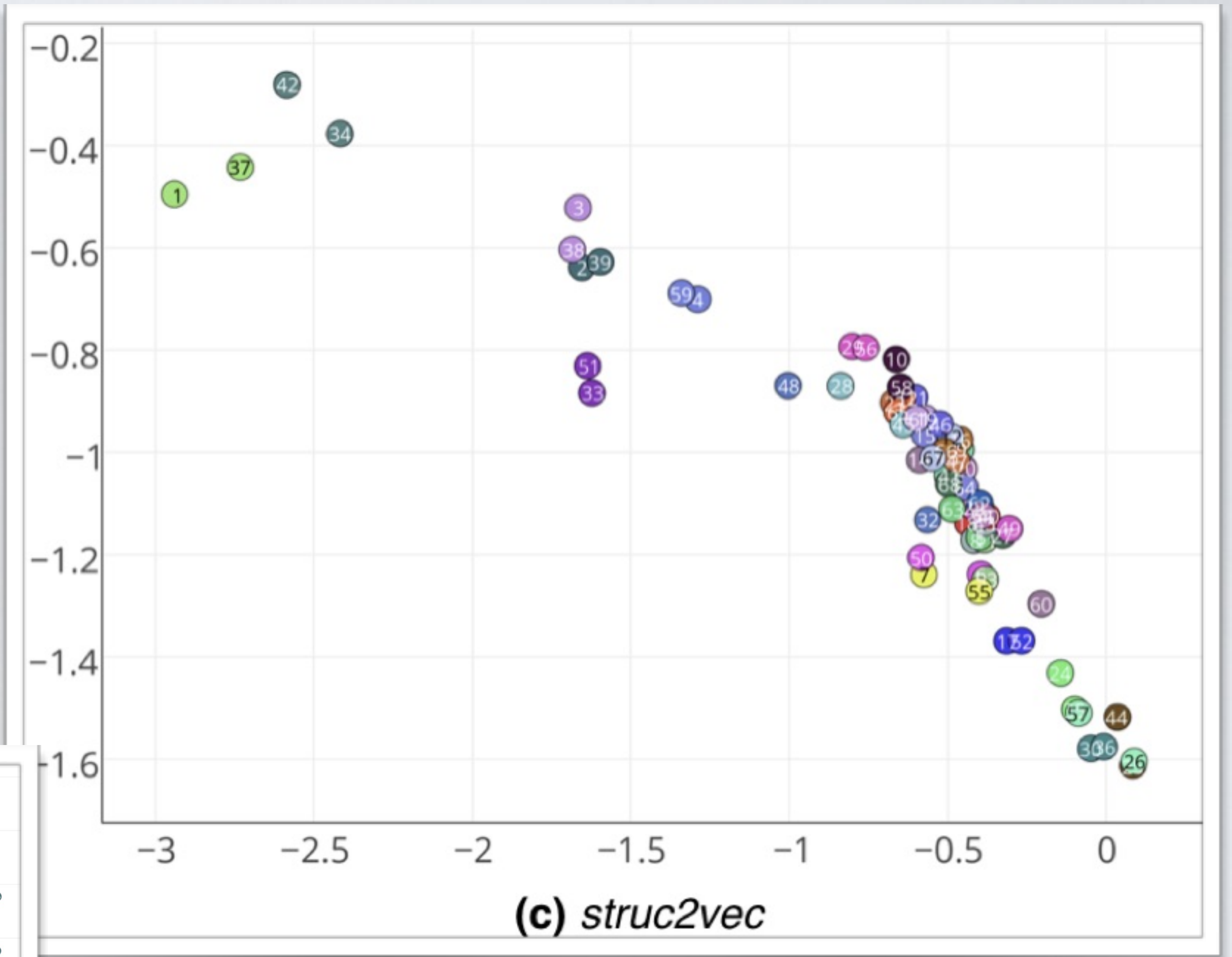
Simple main idea:
Minimize difference between
Weight and cosine similarity

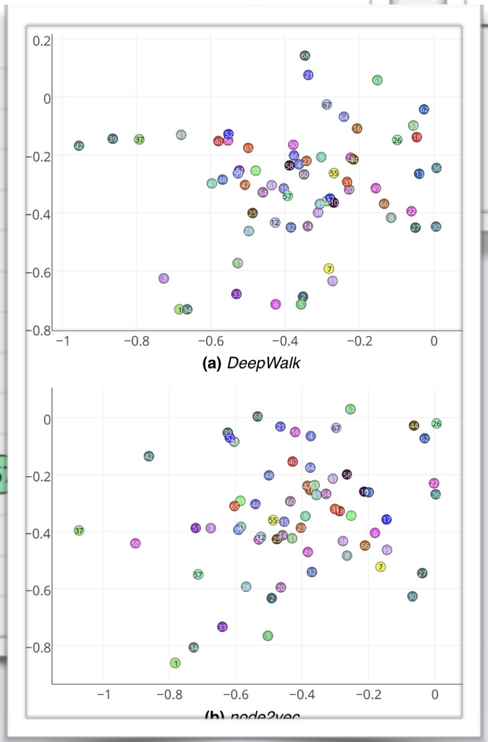
STRUCT2VEC

- In node2vec/Deepwalk, the context collected by RW contain the **labels** of encountered nodes
- Instead, we could memorize the **properties** of the nodes: attributes if available, or computed attributes (degrees, CC, …)
- =>Nodes with a same context will be nodes in a same "position" in the graph
- =>Capture the role of nodes instead of proximity

STRUCT2VEC : DOUBLE ZKC







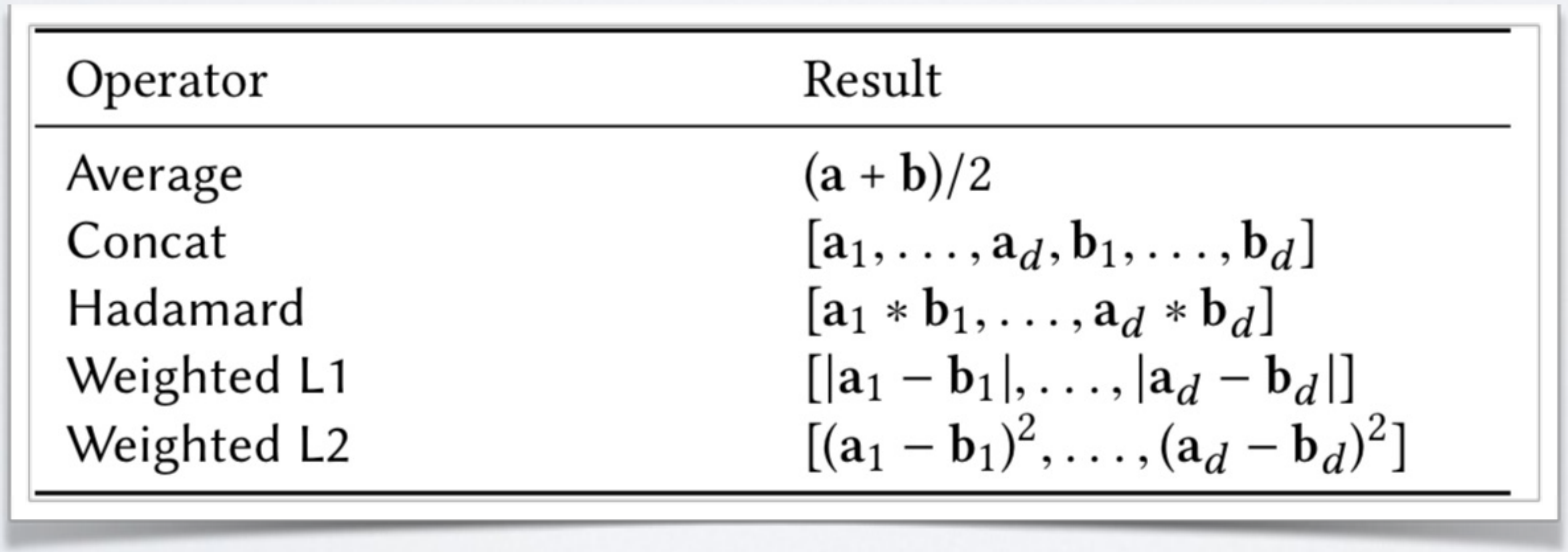
USING EMBEDDINGS

MACHINE LEARNING

- From a set of **features**
  - House: # rooms, surface, volume, number of windows, etc.
- Predict a **value** or a **class**
  - Value: Price, construction year, …
  - Class: Energy efficiency, individual house YES/NO, …
- Using your favorite tool
  - Linear regression, logistic classifier, decision tree, neural networks…

MACHINE LEARNING

- Graph embedding gives you **features** to represent the structural position of **a node**
- Features can be combined to represent edges



Operator	Result
Average	$(\mathbf{a} + \mathbf{b})/2$
Concat	$[\mathbf{a}_1, \ldots, \mathbf{a}_d, \mathbf{b}_1, \ldots, \mathbf{b}_d]$
Hadamard	$[\mathbf{a}_1 * \mathbf{b}_1, \ldots, \mathbf{a}_d * \mathbf{b}_d]$
Weighted L1	$[\lvert\mathbf{a}_1 - \mathbf{b}_1\rvert, \ldots, \lvert\mathbf{a}_d - \mathbf{b}_d\rvert]$
Weighted L2	$[(\mathbf{a}_1 - \mathbf{b}_1)^2, \ldots, (\mathbf{a}_d - \mathbf{b}_d)^2]$

MACHINE LEARNING

- You can predict things about nodes, edges, or pairs of nodes (with/without edge)
- You can combine the embedding information with other features
- =>Use embedding + road characteristic to predict the probability to have congestion, critical failure, speed limit…
- You can also combine with network properties (centralities…)