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
- Modularity counts **weights** inside/between communities
- Sone notions are harder: Clustering Coefficient? Graph Distance?

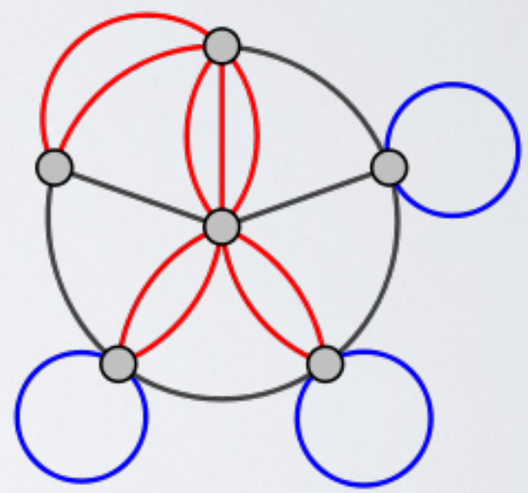
DIRECTIONS

- **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

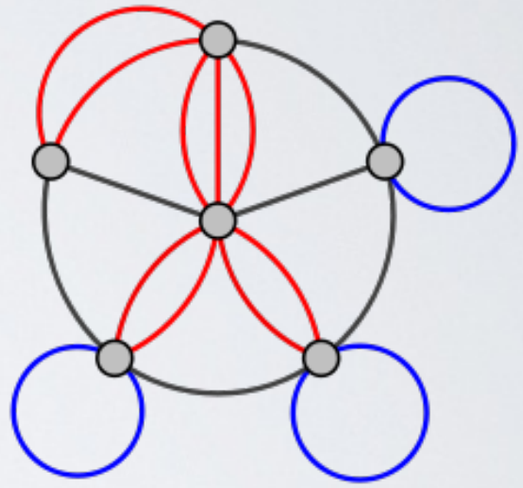
- Multi-graph: several edges allowed between same nodes
- Often used in conjunction with labels:
  - One edge for friendship,
  - one edge for family,
  - one edge for co-worker…



- Without labels, can be simplified as a weighted graph

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
- If several labels, can be considered as separate graphs for analysis

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
  - Reserchers - conferences/institutions
  - …
- Normal methods work but sometimes give unintuitive results:
  - Specific variants have been proposed

MULTI-PARTITE GRAPHS

Modularity: do not count pairs of nodes of same types

$$Q_{\mathrm{B}} = \frac{1}{m}\sum_{u=1}^{r}\sum_{v=1}^{c}(\tilde{A}_{uv} - P_{uv})\delta(g_u, h_v) = \frac{1}{m}\sum_{u=1}^{r}\sum_{v=1}^{c}\left(\tilde{A}_{uv} - \frac{k_u d_v}{m}\right)\delta(g_u, h_v),$$

MULTI-PARTITE GRAPHS

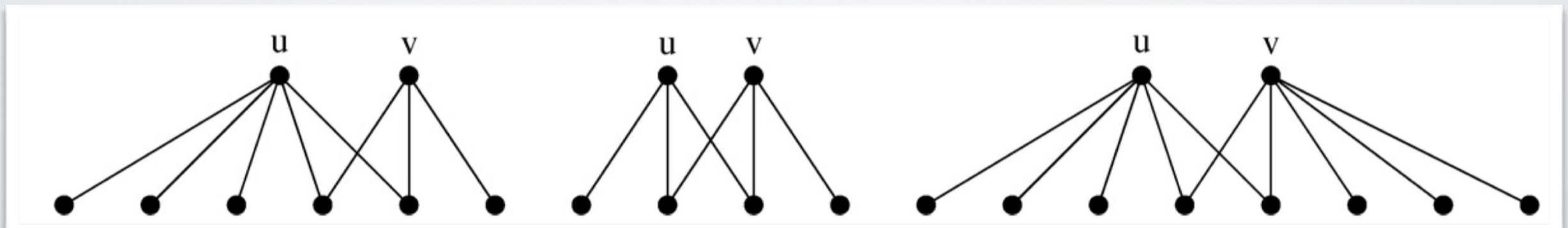
Clustering Coefficient: (one definition)

Of a pair

$$\mathrm{cc}_\bullet(u, v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

Average among N at dist=2

$$\mathrm{cc}_\bullet(u) = \frac{\sum_{v \in N(N(u))} \mathrm{cc}_\bullet(u, v)}{|N(N(u))|}$$



2/6 2/4 2/8

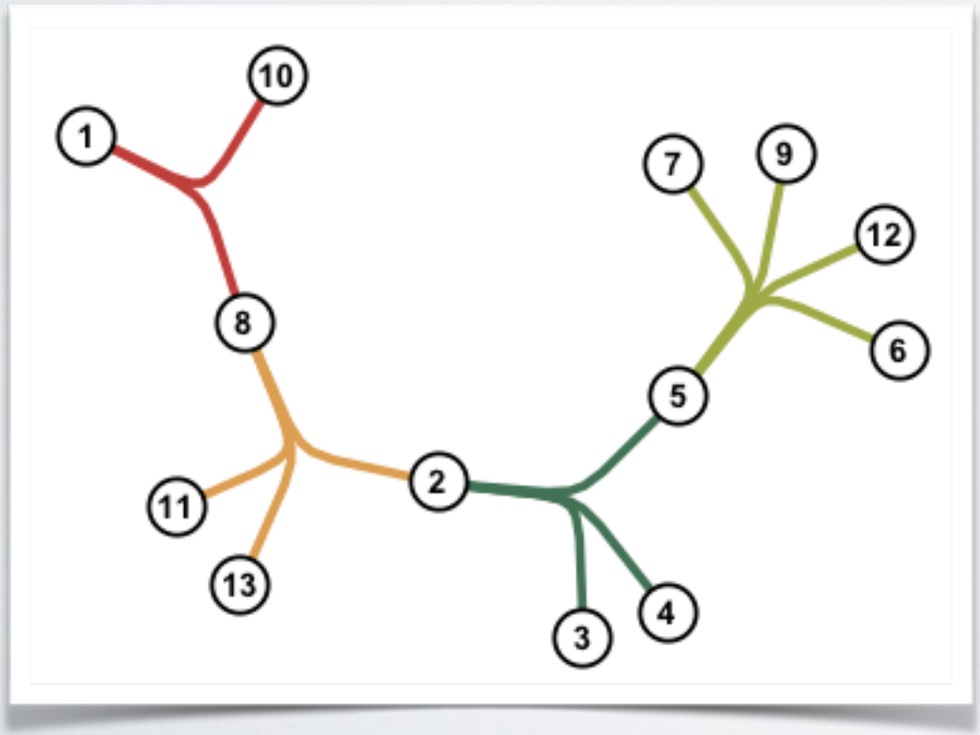
MULTI-PARTITE GRAPHS

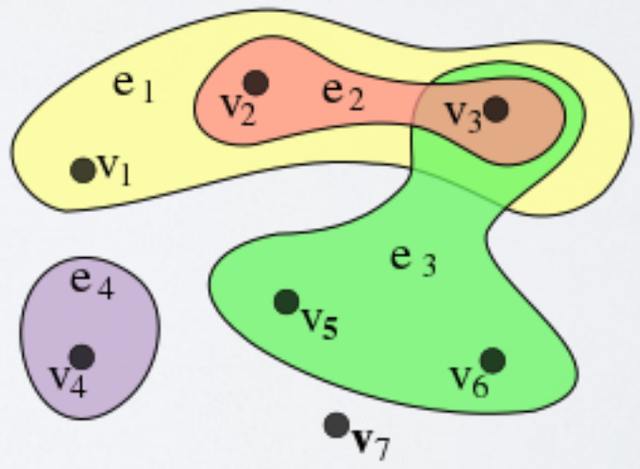
- If you are interested, there is a lot of literature on the topic, in particular domain of **recommendation**
  - Users - products => propose the right products to the right user

HYPERGRAPHS

HYPERGRAPHS

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





HYPERGRAPHS

- Most common usage: represent a single event involving several nodes

- In social networks: 10 students attending a same course A
  - Normal network: 45 undirected edges. Giant clique. Very dense
    - Problem: if 5 attend another course B and the others another course C => no way to see who worked with whom (a single clique, with double links or weights=2)
  - Hypergraph: A single link with ten endpoints
    - And we can add 2 single links with 5 endpoints and still differentiate attendances

- Another example: in Bitcoin, transactions are multi-input, multi-output. Some transactions have 1000 input, 1000 output
  - 500 000 links for a single transaction in a normal network!

HYPERGRAPHS

- In practice, very few direct usages
  - Too difficult to handle ? To different from normal networks?

- Hypergraphs can be transformed in bi-partite graphs
  - Social Network: student nodes and classes nodes
  - Bitcoin: transaction nodes and addresses nodes

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
    - ...

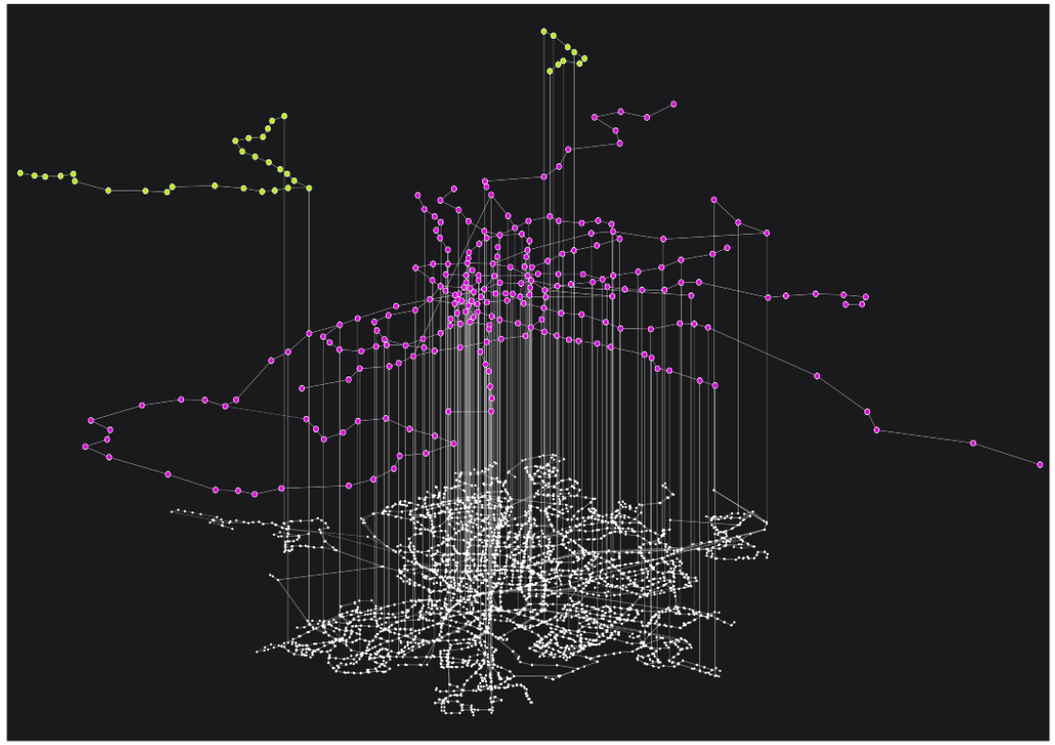
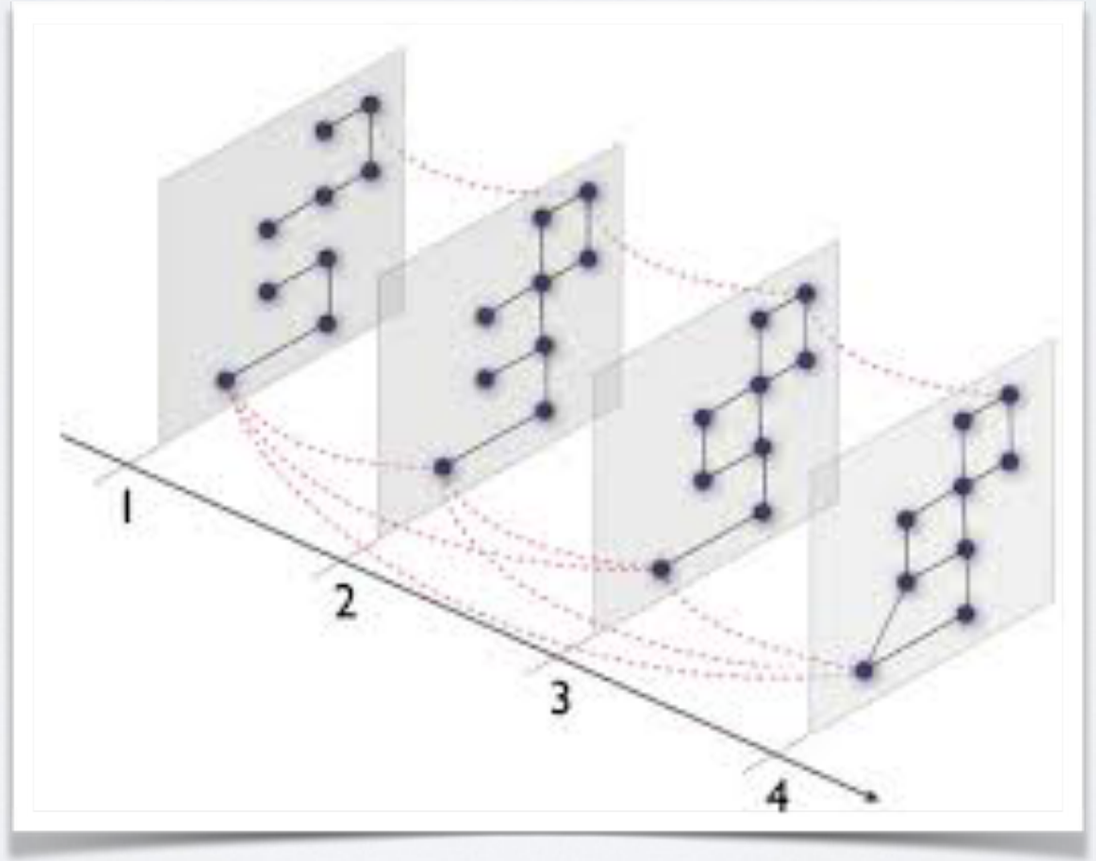


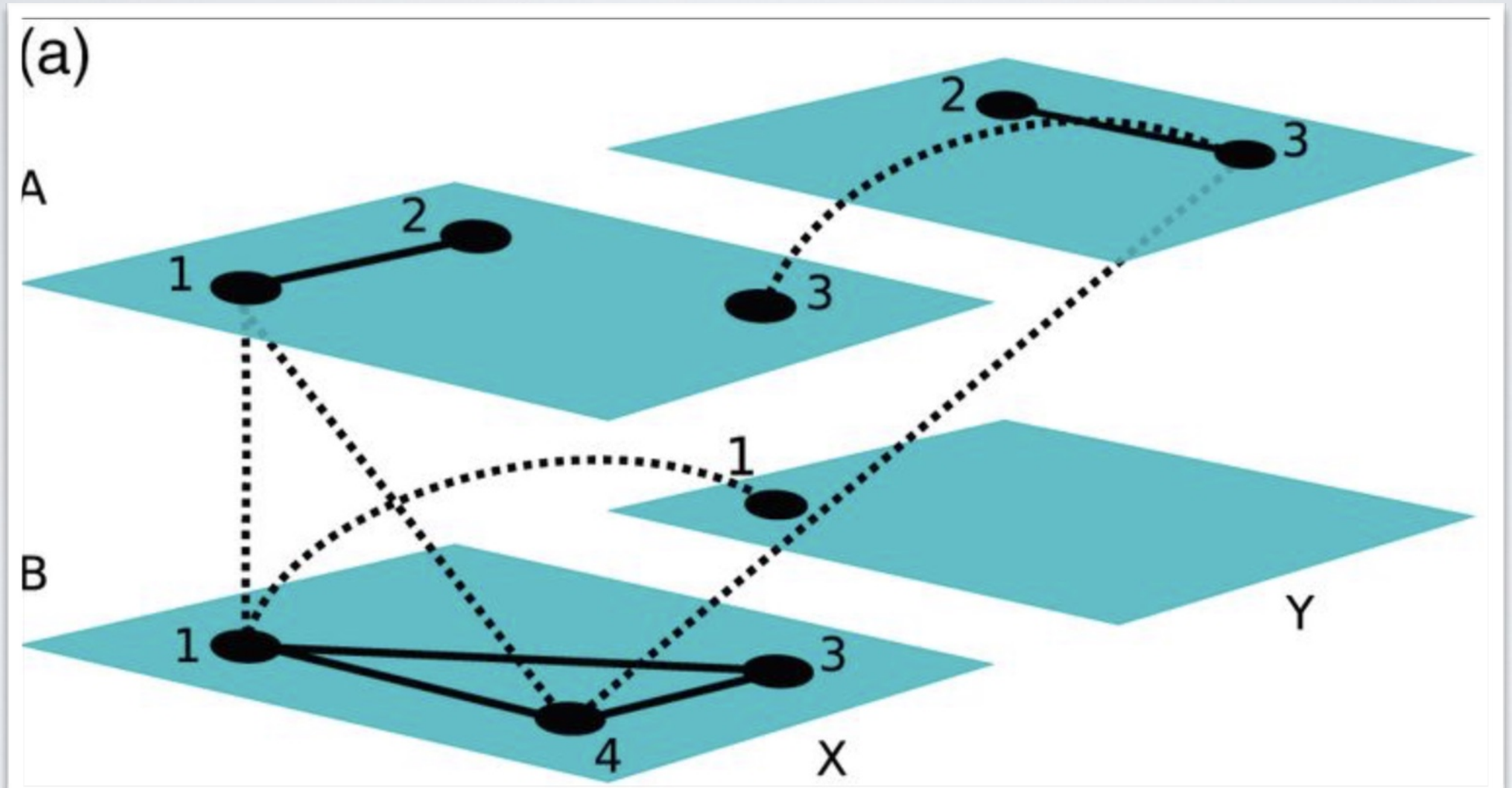
Figure 2. Superlayer representation of the Madrid transportation system. The figure represents the three transportation modes considered: tram (yellow nodes, upper layer), metro (purple nodes, mid layer) and buses (white nodes, bottom layer). See Table1 for statistics of these layers.

MULTILAYER NETWORKS

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



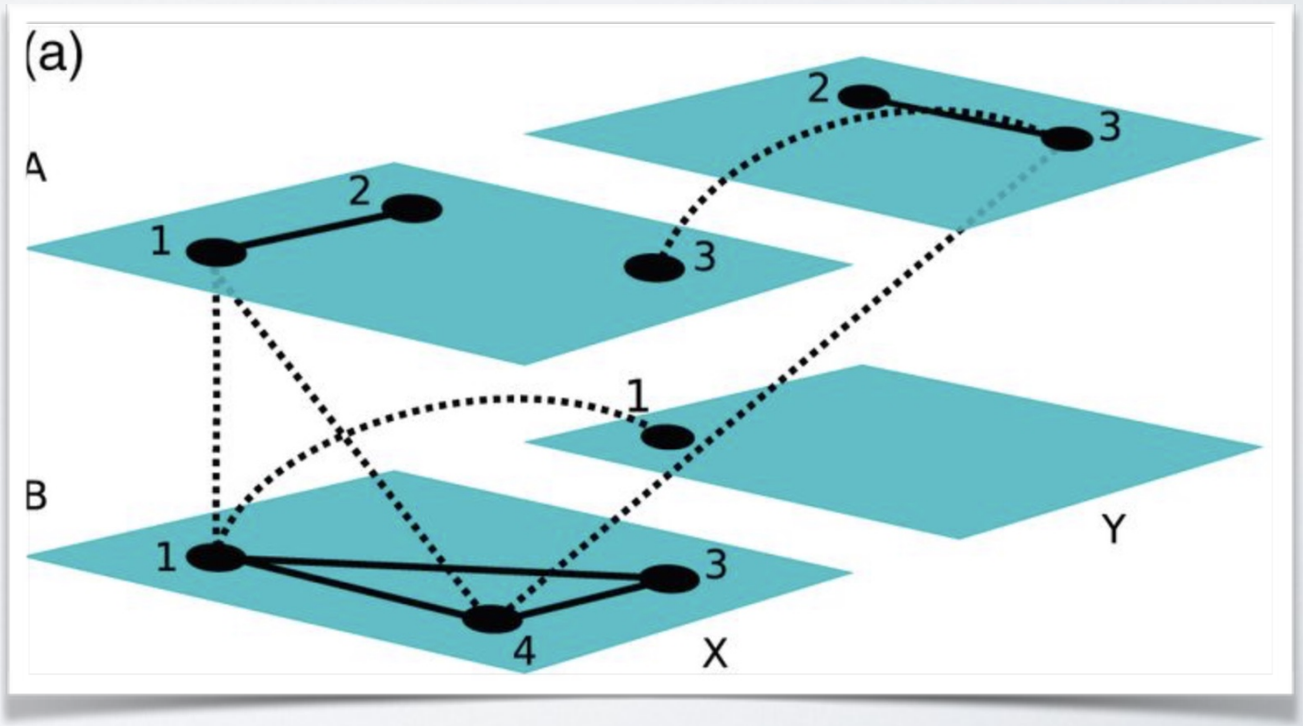
MULTILAYER NETWORKS



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.

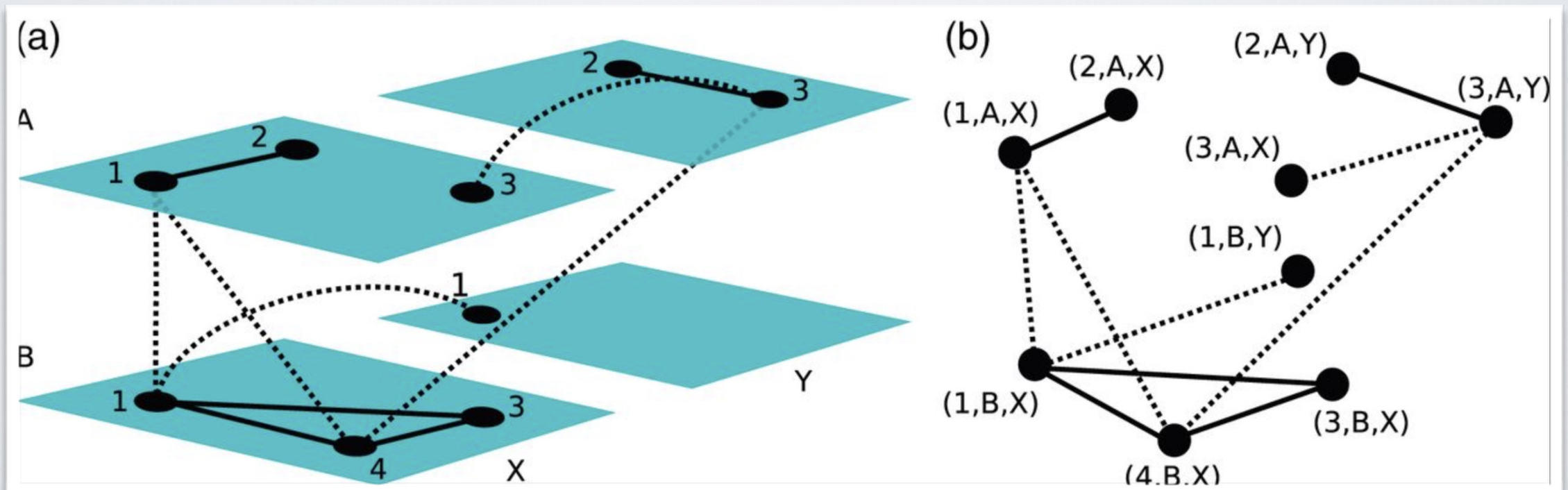


MULTILAYER NETWORKS

- 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

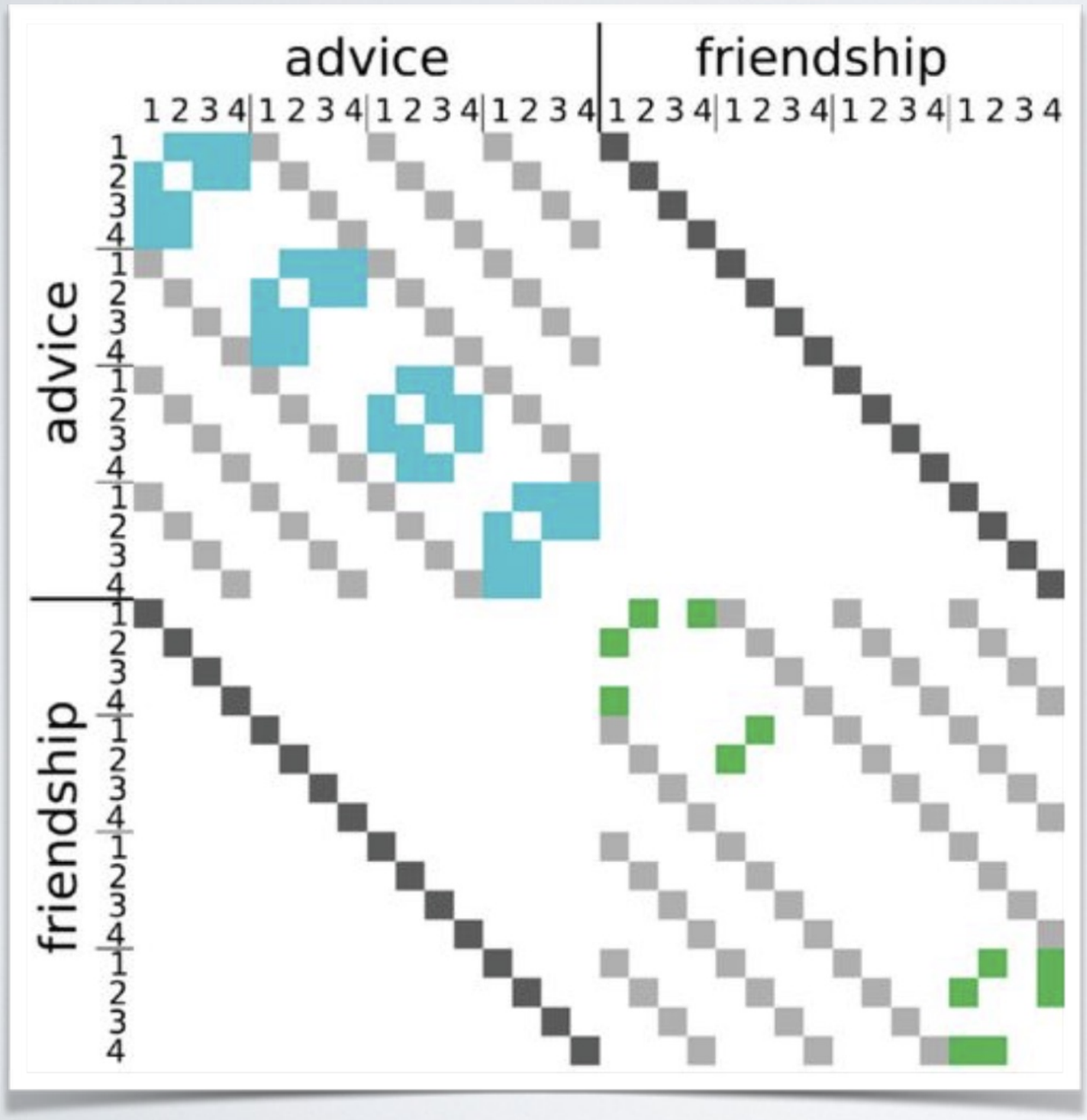
A simple idea: multilayers networks can be transformed into normal networks



MULTILAYER NETWORKS

- Matrix representation:
  - Many algorithms on networks work on adjacency matrices
- Solution 1: Tensors
  - Be careful if not all nodes in all layers!
  - Interesting only when only links between same nodes in <> layers
- Solution 2: Supra-adjacency matrix
  - Or flattened tensors

MULTILAYER NETWORKS



Blue, green: intra-layer

gray: inter-layer 1

black: inter-layer2

Cognitive map: relations between
4 people seen by each of these 4 people

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.

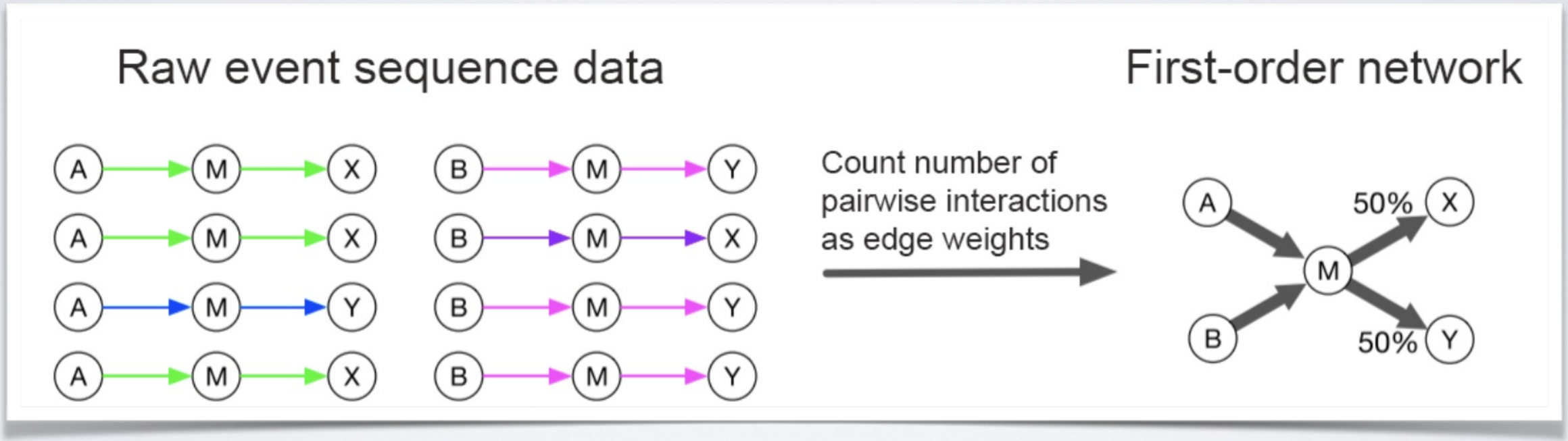
- Many cities does not have direct trips
  - e.g.: Paris->Cali
  - Flight goes through stopover: Paris->Bogota->Cali

- If we want to create a weighted network of trips:
  - We count the number of trips Paris->Bogota, and Bogota->Cali
  - We forget the information of previous/Next step

HIGHER ORDER NETWORKS

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

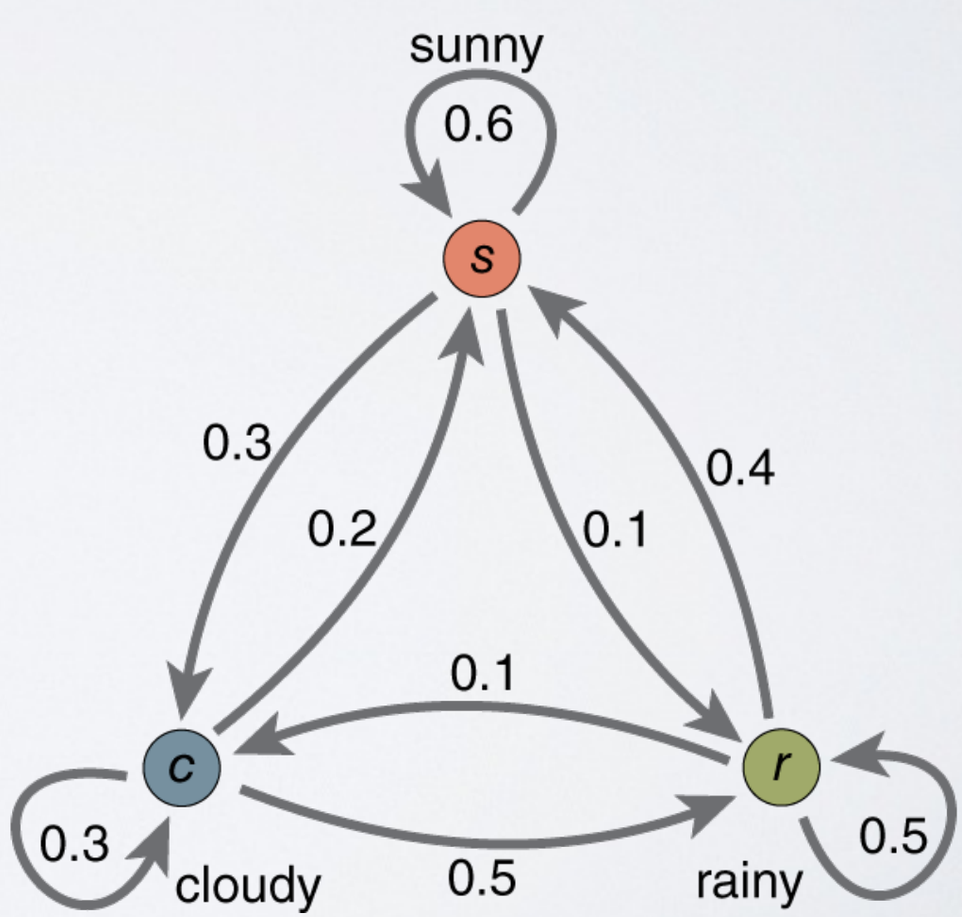
HIGHER ORDER NETWORKS



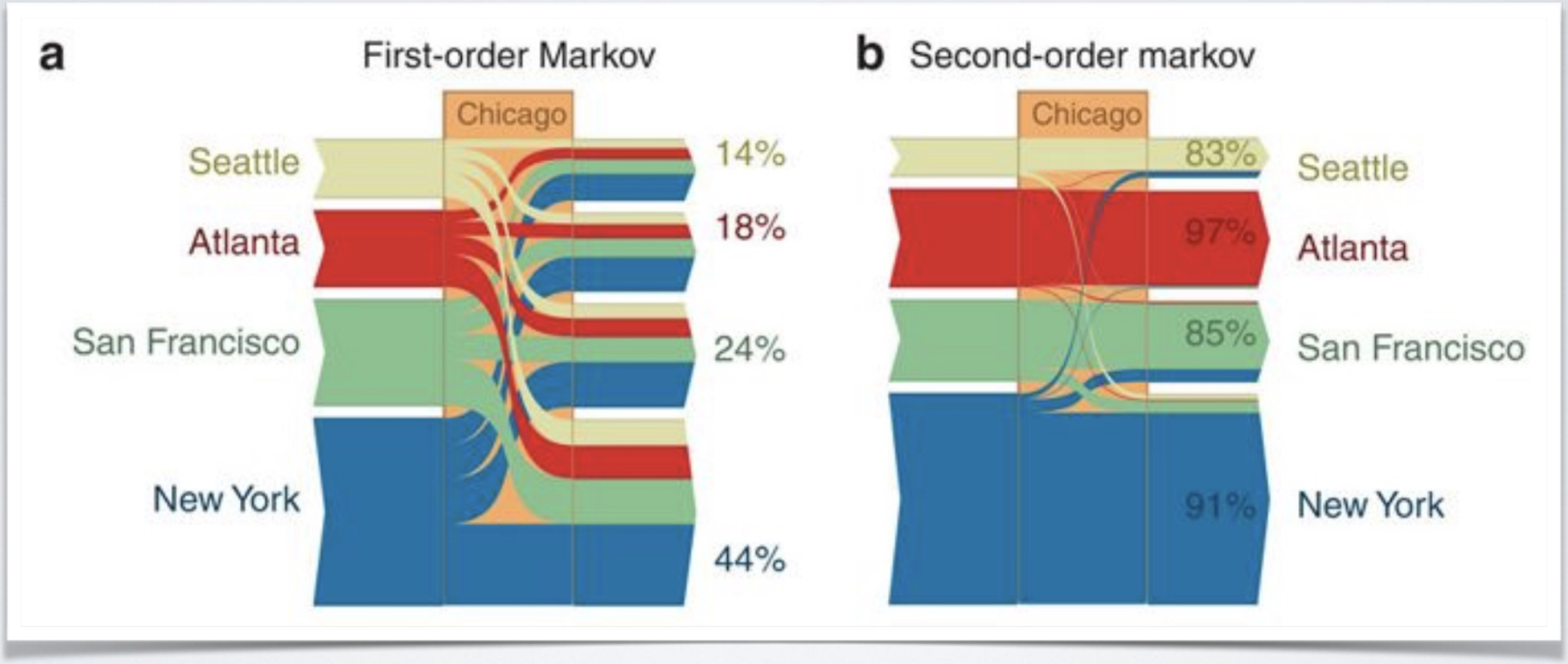
[Xu 2016]

HIGHER ORDER NETWORKS

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



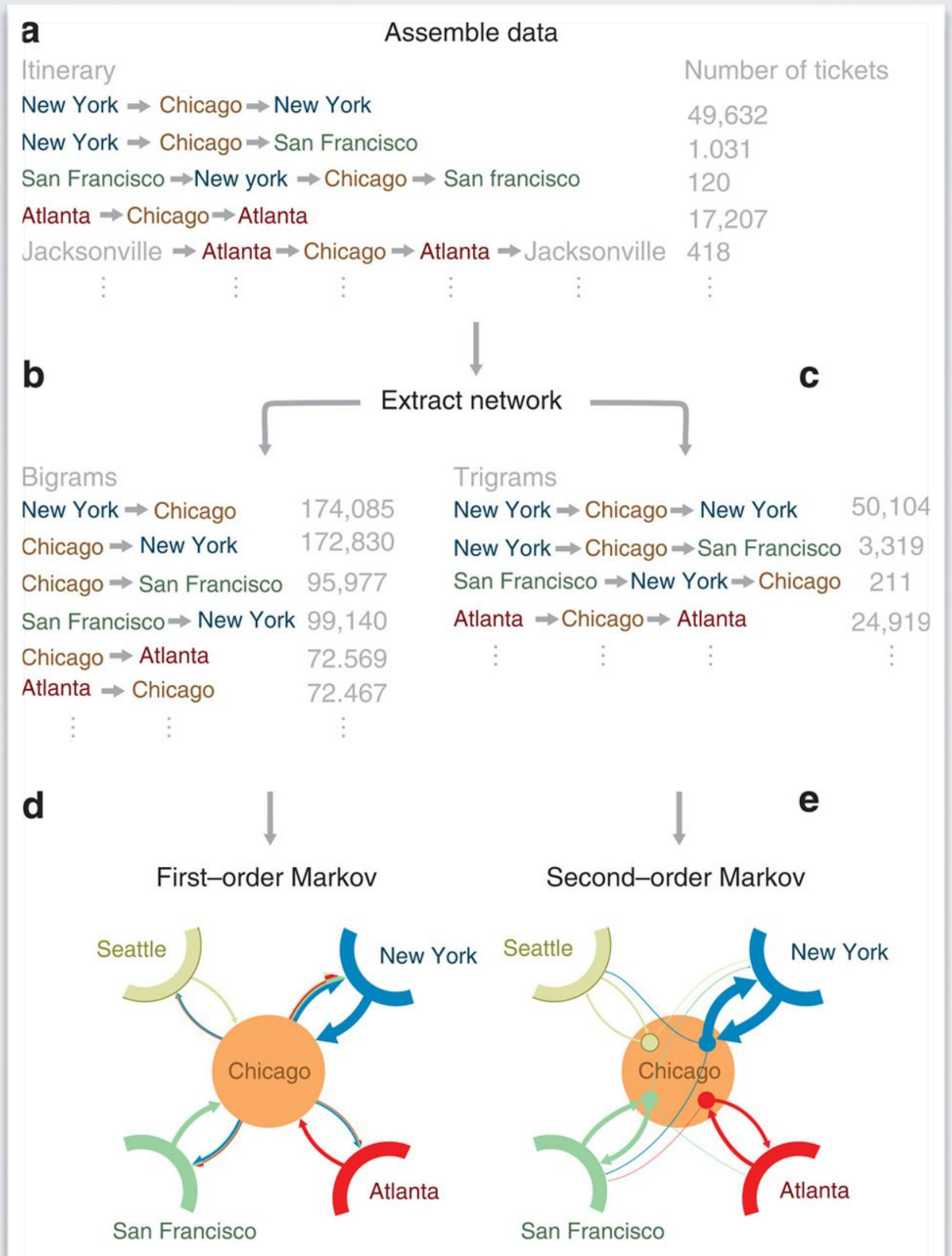
HIGHER ORDER NETWORKS



Round trips

[Rosvall 2014]

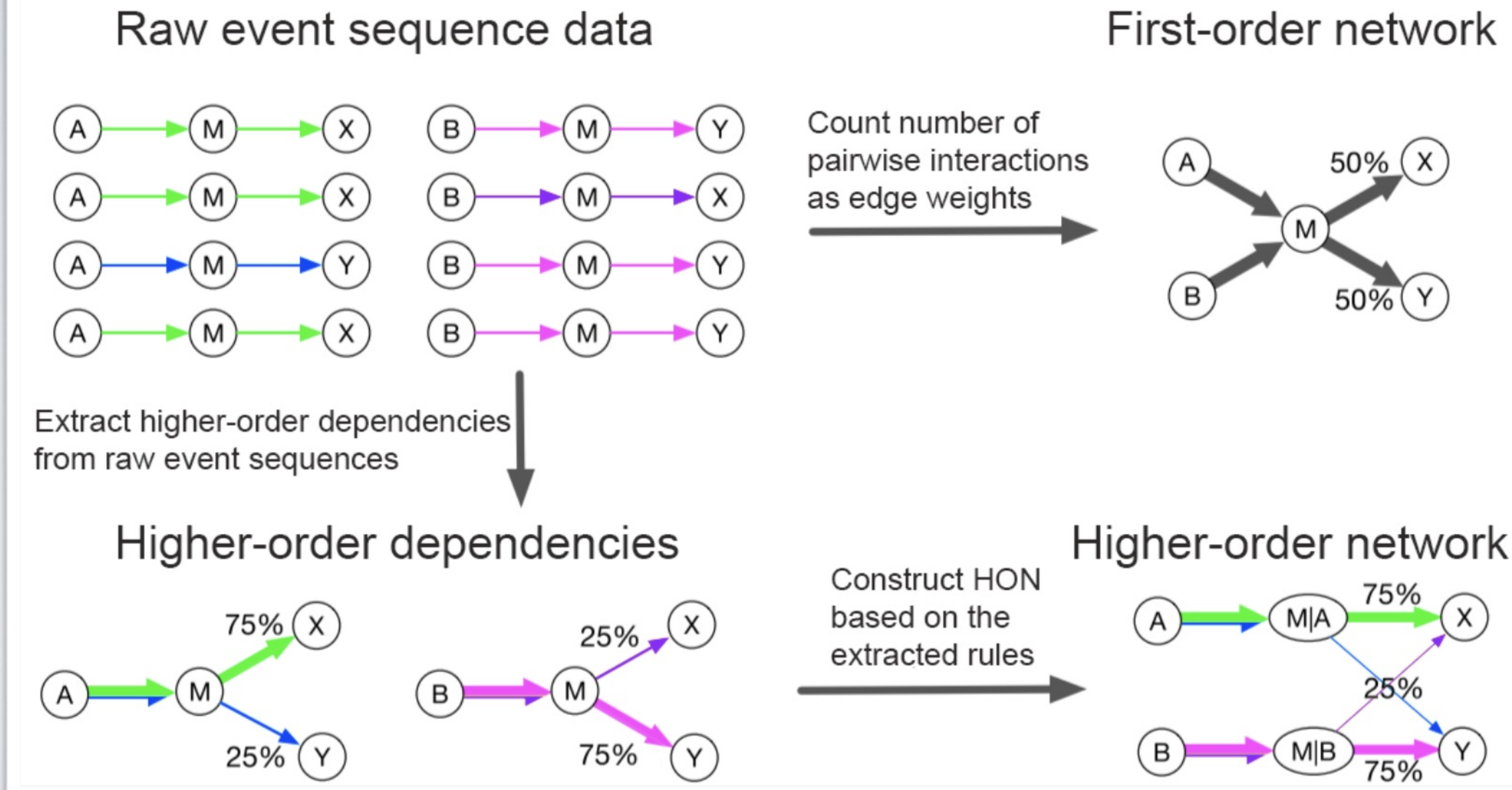
HIGHER ORDER NETWORKS



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



HIGHER ORDER NETWORKS

- Do you remember the **random walk** approches?
  - Centrality: PageRank
  - Communities: Infomap

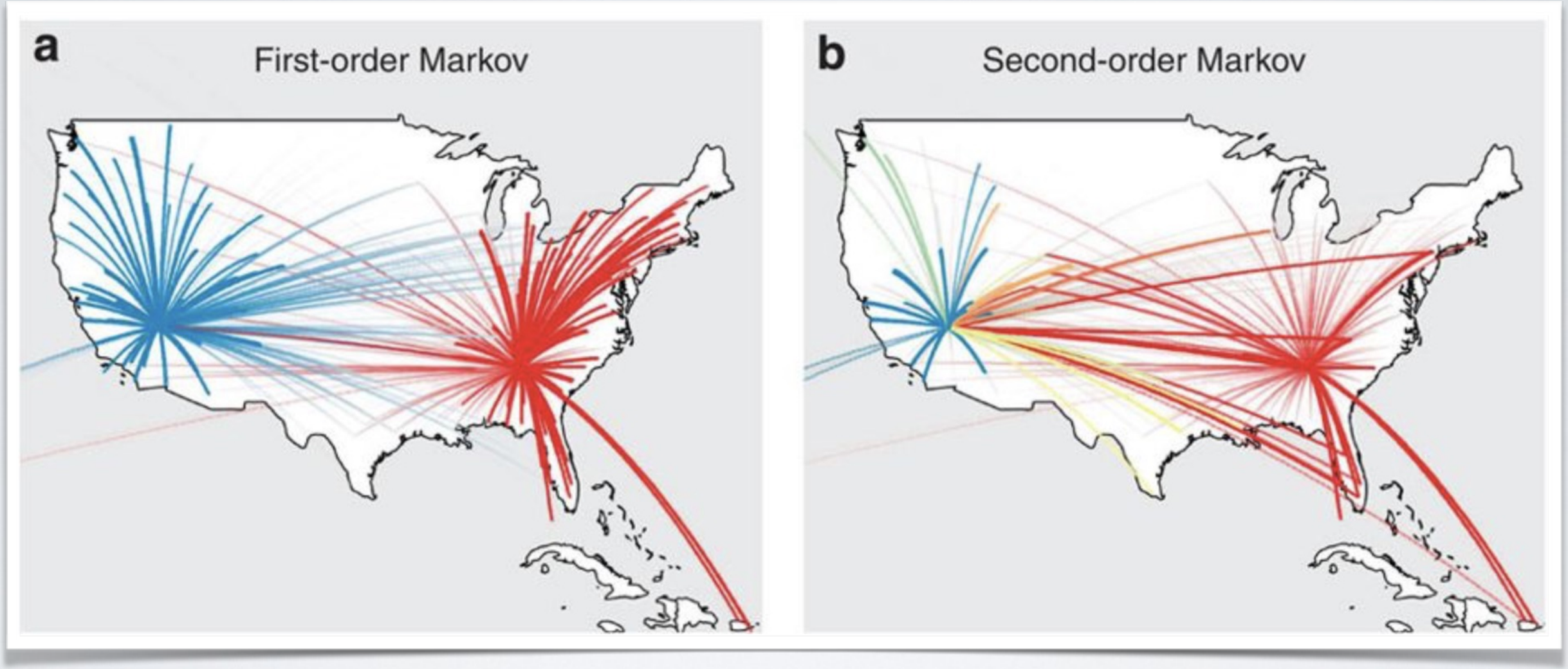
- They are based on the principle of a markovian processes
  - At each step, the *random walker* decides to follow an out-going link
  - Same probability to go back from when it comes than go forward!
    - Normal, without other knowledge (air traffic<>browsing the web)

- We can generalize them to higher order naturally

HIGHER ORDER NETWORKS

- InfoMap: compress the walk of a higher order random-walk
- Remember: a node is a tuple (node,history)
- If we apply a community detection algorithm:
  - Communities are composed of (node,history)
  - We can go back to a traditional community partition:
    - We forget the memory part of nodes
    - Several instances of same nodes in same community
    - Same node in different communities
    - =>Overlapping communities

HIGHER ORDER NETWORKS



Atlanta: hub platform (triads)
Las Vegas: touristic place (round trips)

[Rosvall 2014]

HIGHER ORDER NETWORKS

- PageRank:
  - Score of node A: Sum of the pagerank of all instances of node A

- Introduce large change in rankings

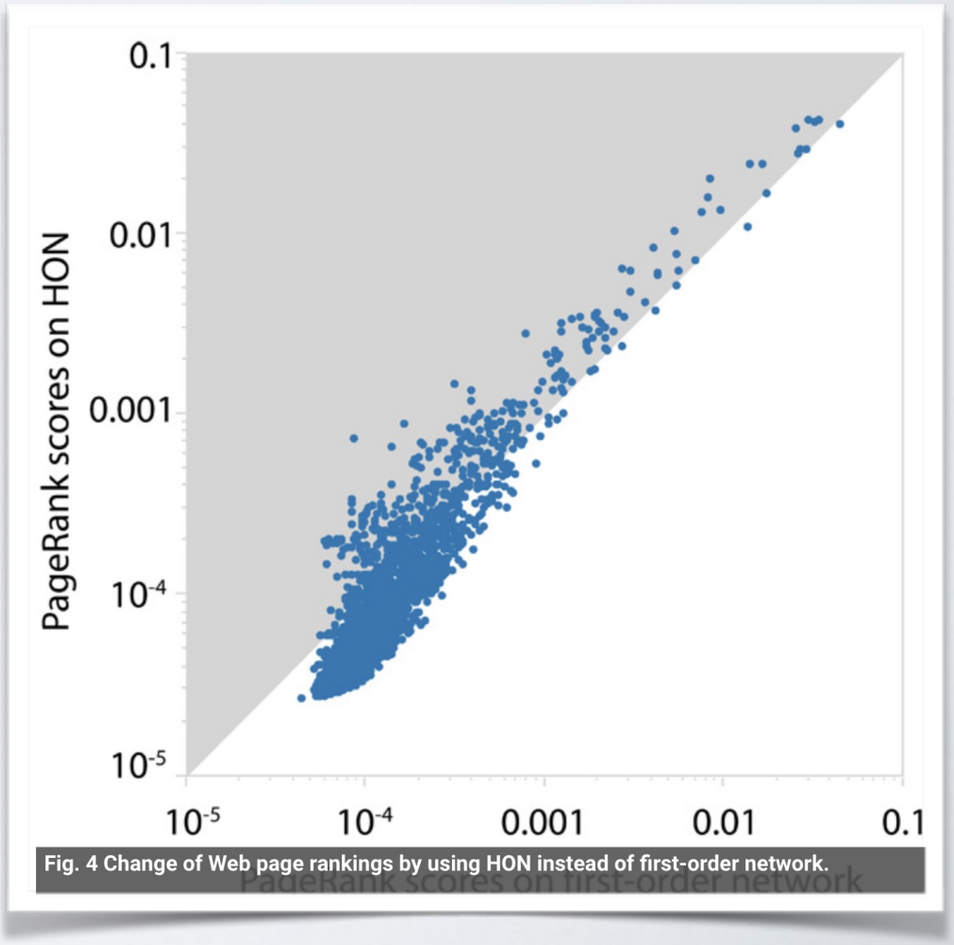
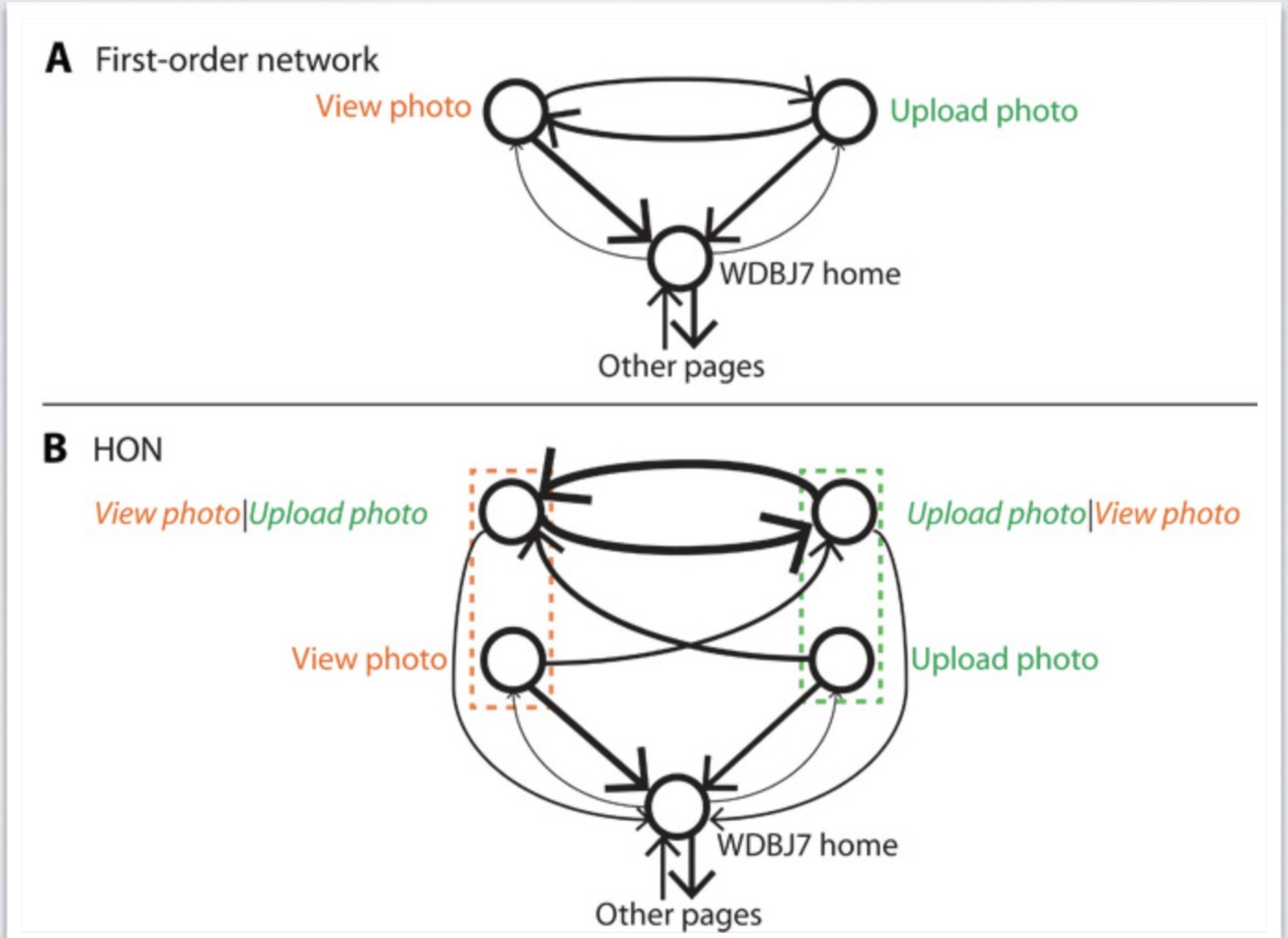


Fig. 4 Change of Web page rankings by using HON instead of first-order network.

HIGHER ORDER NETWORKS



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

PRACTICALS

PRACTICALS

- 1)Download the data about Bogota bus routes
- 2)Create the FIRST order network
  - Nodes are stations
  - Several possibilities for edges:
    - Edge only between successive stations (too limited ?)
    - Edge between all stations of a same line (too dense for your computer?)
    - Edge between stations at less than 3km (a little more code)
    - First remove all stops with less than k=4 different lines, then edge between all stations of a same line
  - If several times the same edge: increase weight
- 3)Create the SECOND order network
  - Be careful at the explosion of complexity!

PRACTICALS

- 4) Compare PageRank for first and second order network
  - Don't forget about weights! (look at the weight option of the pagerank function)
- 5)Compare communities for first and second order networks,
  - in particular, identify overlapping nodes