Statistical Network Analysis

Prof. Dr. Ingo Scholtes

Chair of Machine Learning for Complex Networks
Center for Artificial Intelligence and Data Science (CAIDAS)
Julius-Maximilians-Universität Würzburg
Würzburg, Germany

ingo.scholtes@uni-wuerzburg.de

Lecture 13
Causal Topologies in Temporal Networks

February 1, 2023

Notes:

- **Lecture L13:** Causal Topologies in Temporal Networks 01.02.2023
- **Educational objective:** In the final lecture of this course, students will understand why patterns in time series data add an additional dimension of complexity that must be considered in network analysis and the modelling of dynamical processes in networks.
  - The arrow of time in networks
  - Higher-order network models and diffusion speed
  - Optimal higher-order models

- **Reminder**: There is no exercise sheet released this week. Next week's lecture will be a **repetitorium**. I will bring all lecture slides and exercise solutions and will repeat topics/slides on a request basis. Please prepare your questions.

- **Final Exam**: The exam is scheduled for **February 15th 2023, 12:00 - 15:00**. It will be a written exam of 120 minutes.

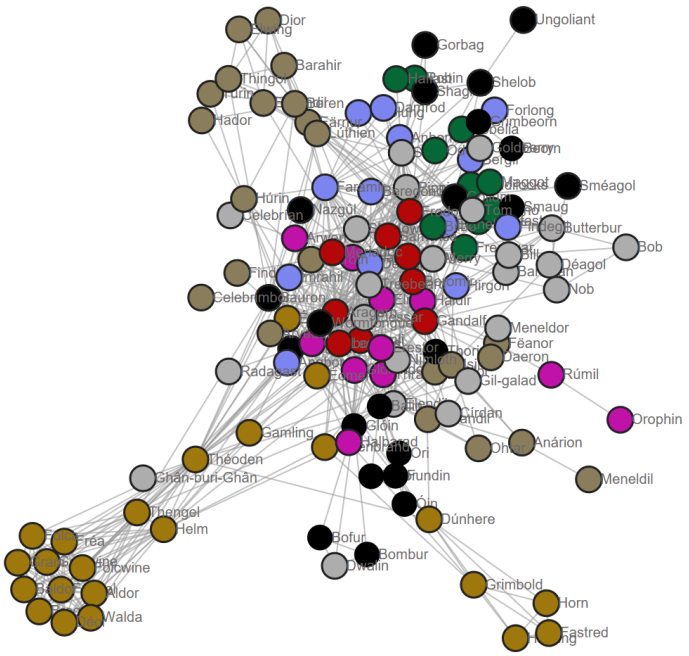
Motivation

- we explored **random walks and diffusion in networks**
- important foundation for network analysis and **machine learning in networks**

  → Course: Machine Learning for Networks, summer 2022

exemplary network analysis techniques

- random walk centrality, PageRank
- eigenvector centrality
- spectral clustering
- flow compression (InfoMap)
- graph embedding (node2vec, DeepWalk)



static network of character co-occurrences in “The Lord of the Rings”

- throughout our course, **we assumed that networks are static**!

Notes:

- In the last three lectures, we explored random walks and diffusion in networks, an important class of dynamical processes that serve as theoretical foundation for data mining and network analysis techniques.
- We can, for instance, use random walks to detect “natural groups” of nodes by studying sets of nodes within which a random walker gets trapped for extended periods of time or we can use spectral partitioning to detect natural clusters. We can further assess the importance of nodes based on the frequency by which they are visited by a random walker (see e.g. PageRank). Moreover, random walks are the basis for popular graph embedding techniques, which embed nodes into a vector space that can be used as feature space for downstream machine learning techniques. We will cover such methods in more detail in our upcoming course *Machine Learning for Complex Networks*, which will be held in the summer semester.
- While we have considered dynamical processes on networks, throughout our course, we have assumed that the network itsrelf is static, i.e. all links are present all the time.
- In today’s lecture, we show which additional challenges are introduced when modelling dynamical processes in dynamic networks, i.e in networks where links change over time.

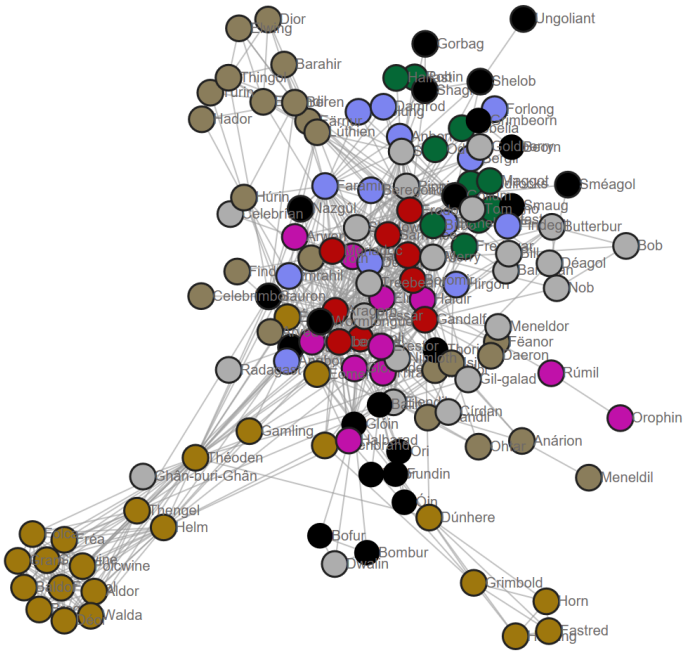
Networks and time

problem

in many real data sets on networks **interactions/relationships carry time stamps**, e.g.

- time-stamped (social) interactions
- financial transactions
- protein-protein interactions
- click streams of Web users
- temporal co-occurrence

character_a	character_b	sentence
Gandalf	Frodo	245
Frodo	Sam	323
Sam	Meriadoc	420
Frodo	Meriadoc	478
Meriadoc	Peregrin	515
Gandalf	Sam	766
...	...	...



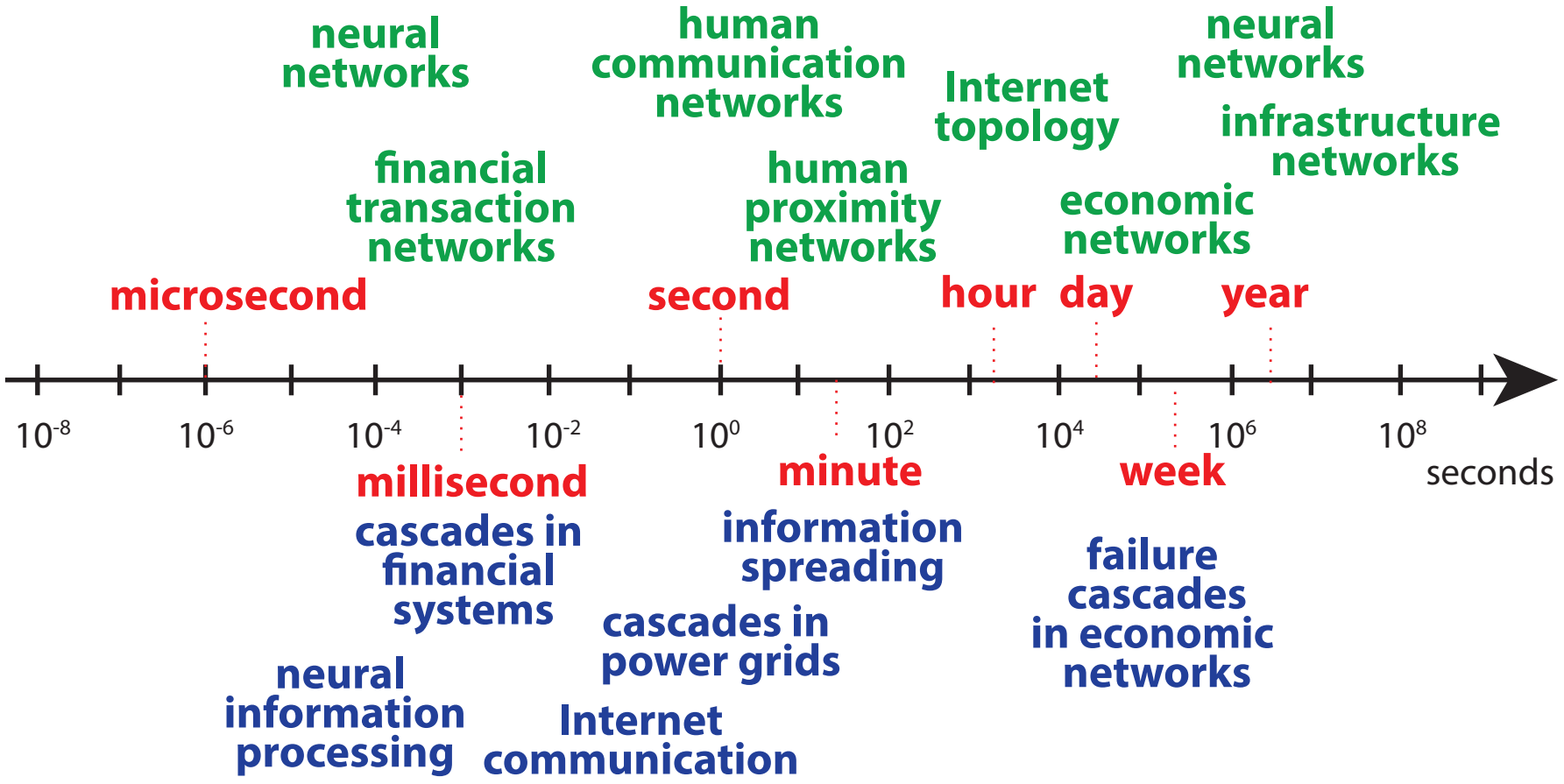
static projection of
dynamic character co-occurrences
in "The Lord of the Rings"

- how does the **dynamics of networks** influence dynamical processes?

Notes:

- In many real network analysis tasks, we have access to *time series data*, i.e., data that not only tell us who is connected to whom, but also *when* (and in which order) interactions or relationships occur. In fact, many of the network data that we have used throughout the course included temporal information, which we have ignored so far for simplicity. That is, we have considered "time-aggregated" projections where a link exists if it occurs at any time during the observation period. Naturally, with this perspective we lose information on the underlying system.
- As a simple example, consider character co-occurrences in the novel "The Lord of the Rings". Here we consider an "interaction" between two characters to exist whenever they are mentioned in the same sentence, and we consider the number of this sentence to be the time stamp. We obtain a so-called *dynamic* or *temporal network*, i.e., a network where we know the time and/or sequence of links. In the example, we see that the visualisation of such a temporal network gives us clues about the actual story that we cannot get from a static network visualisation.
- The question how we can analyse such time series data is challenging. Considering that an understanding of dynamical processes in static networks is the basis for data mining and machine learning, here we show how we can model dynamical processes like, e.g., diffusion in a dynamic network and how we can use such models as a basis for network analysis and machine learning. Much of our group' research in the past five years (and probably the next five years) is devoted to this question. In today's lecture, we demonstrate some of the challenges arising in time series data on networks, and how to address them based on **higher-order network models**.

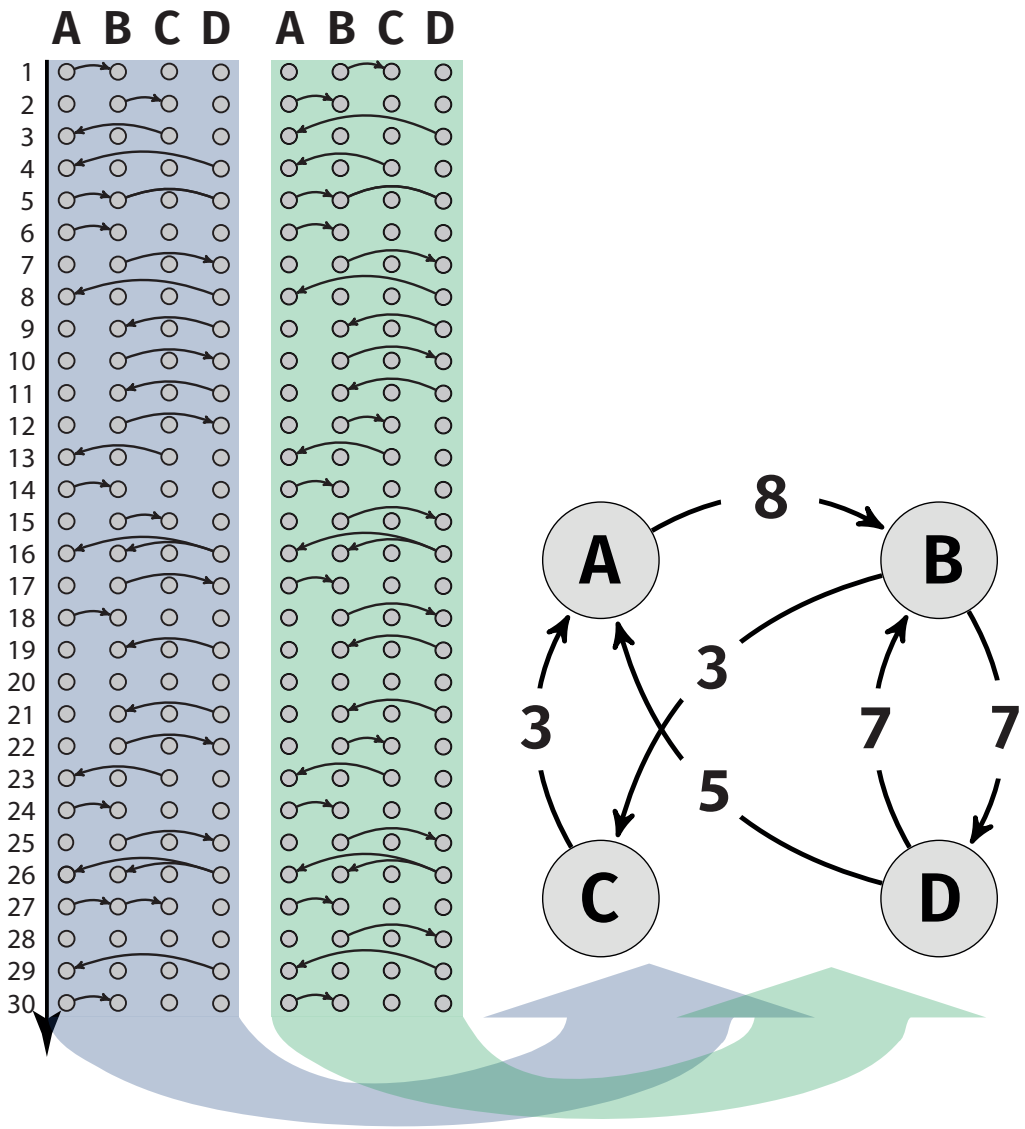
Time scales: networks vs. processes



Notes:

- As explained in lecture 10, we are often not interested – per se – in the network topology of a system but rather in the dynamical processes that define a system's "function". Examples for processes of interest are diffusion dynamics, information propagation, routing of data packages in communication networks, or cascading failures propagating within financial systems.
- If and how the dynamics of networks influence such processes depends on the time scales at which the network topology and the process evolve. If the network evolution is much slower than the evolution of the process, the network appears to be static from the perspective of the process. In such cases, we can often ignore the network dynamics in the description of the process. If, on the other hand, the evolution of the network occurs on a time scale that is much faster than the time scale at which the dynamical process evolves, then the details of this network dynamics may actually not influence the process as well (because the slow process effectively sees an "aggregation" of the many dynamic changes occurring in the network).
- As a rule of thumb, **it is particularly important to consider network dynamics when the time scales at which the process and the network evolve cannot be separated**. This is, for example, the case in information spreading in human communication networks. Here, the network dynamics can actually "drive" the evolution of the process, i.e. it does not make sense to study the process on a time-aggregated network.

Temporal networks



temporal network

temporal network $G^{(t)} = (V, E^{(t)})$ consists of

1. **nodes** $v \in V$ and
2. **time-stamped links** $(v, w; t) \in E^{(t)} \subseteq V \times V \times \mathbb{N}$ for discrete time stamps $t \in \mathbb{N}$

time-aggregated network

(static) network $G = (V, E)$ with weights $w(v, w) = |\{t \in \mathbb{N} : (v, w; t) \in E^{(t)}\}|$ is called **time-aggregated network** of $G^{(t)}$

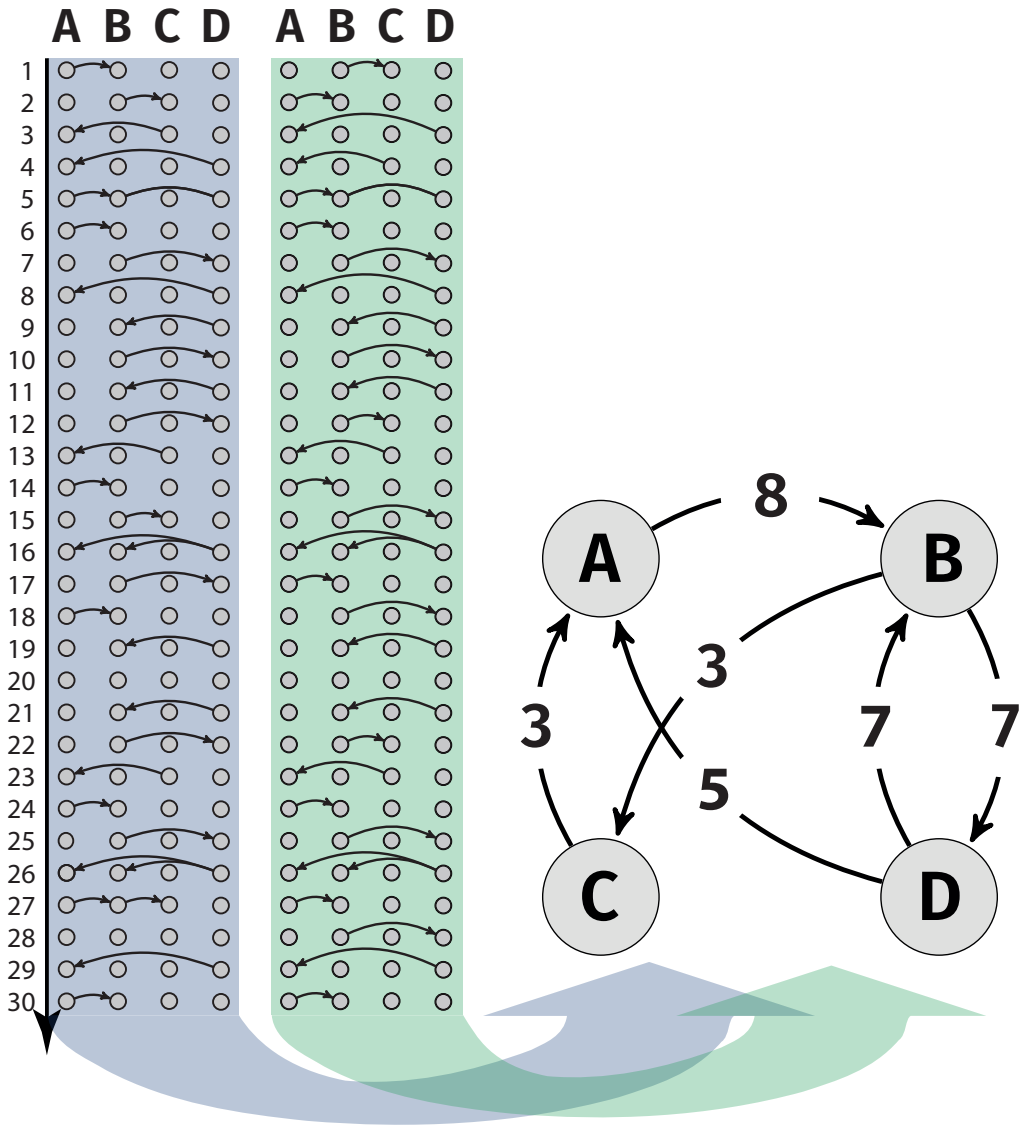
observation

different temporal networks can have the same time-aggregated topology

Notes:

- To understand how network dynamics influences processes, we must first define what exactly we mean by a temporal network, a time-aggregated network, and paths in a temporal network. Indeed, it turns out that the concept of a path in a temporal network is much more complex than what we have learned for static networks.
- Let us consider a **temporal network**, which we define as sequence of time-stamped links that instantaneously occur at discrete time stamps t. In other words: a time-stamped link $(v, w; t)$ means that node v was linked to node w exactly at time t (and the link possibly disappears in the next time step).
- Using this discrete time approach, we can also represent data in which links have a certain *duration*, e.g., by incorporating a maximum sampling frequency Δt, and assuming time-stamped links at times t, $t + \Delta t$, $t + 2\Delta t, \ldots$ over the whole duration of a link.
- We can represent a temporal network by so-called **time-unfolded networks**, a two-dimensional static representation where nodes are arranged on the horizontal axis (i.e. each column represents one node) while time is represented by the vertical axis (i.e. each row represents one discrete time stamp).
- We define a **time-aggregated network** as the projection of such a temporal network onto a weighted, directed, static network, where a link exists whenever two nodes are connected at *any* time. We further assume that the (integer) weight of a link in the time-aggregated network counts the number of times a link has been active.

The arrow of time in networks



causal paths in temporal network

sequence of time-stamped links $(v_0, v_1; t_1), \ldots, (v_{l-1}, v_l; t_l)$ is a **causal path** $v_0 \to \ldots \to v_l$ iff

1. $t_1 < t_2 < \ldots < t_l$ and
2. $0 < t_{i+1} - t_i \leq \delta \; (i = 0, \ldots, l)$

for given **maximum time difference** δ

observation

- chronological ordering of links determines **causal paths**
- ordering can break **transitivity** of paths in networks
- time aggregation **discards information on causal topology**

Notes:

- We now consider the notion of a **path in a temporal network**. We say that a sequence of time-stamped links connects nodes via a **causal path** if the time stamps of these links respect *causality*, i.e. the sequence of time stamps must be ascending. This condition is due to the directedness of the arrow of time. It considerably complicates the study of paths in temporal networks, as opposed to paths in static networks.
- In many situations, we are not interested in causal paths that take an arbitrary amount of time. Consider a data set on time-stamped links occurring between the years 2000 and 2016. If a node a is connected to b in 2000, and b is connected to c in 2016, we may not want a to be considered connected to c via a path that takes 16 years to complete (cf. slide 4).
- We typically include a time scale, i.e. maximum time difference δ after which we do not consider time-stamped links to contribute to a path. In general, the choice of this parameter is non-trivial and it depends on the system/process that we study.
- For illustrative purposes, here we assume a trivial $\delta = 1$, i.e. we only consider two links to form a causal path if they are directly consecutive in time. This makes it easy to see causal paths in the time-unfolded networks shown on the left.
- The simple example illustrates an important fact: Due to the directedness of the arrow of time, the chronological order in which time-stamped links occur determines causal paths. However, this dimension of the data is discarded by a time-aggregated network model. We specifically observe that, different from a time-aggregated network, causal paths in a temporal network are not necessarily transitive.

Diffusion in temporal networks

- graph analytic & algebraic methods: **transitive & Markovian paths**, i.e. $P(c|a \to b) = P(c|b) \cdot P(b|a)$
- correlations in the activation sequence of links lead to **non-Markovian paths** that can **break transitivity**
- invalidates use of matrix powers and spectral methods

misleading results about ...

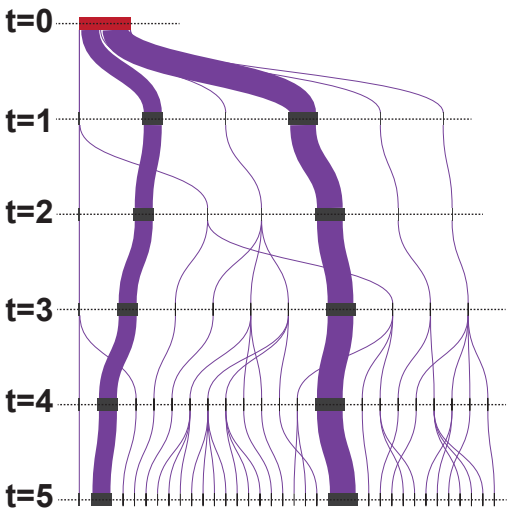
- random walks
- node centralities
- cluster structures

→ R Pfitzner, I Scholtes et al., 2013
→ M Rosvall et al., 2014
→ I Scholtes, 2017

Notes:

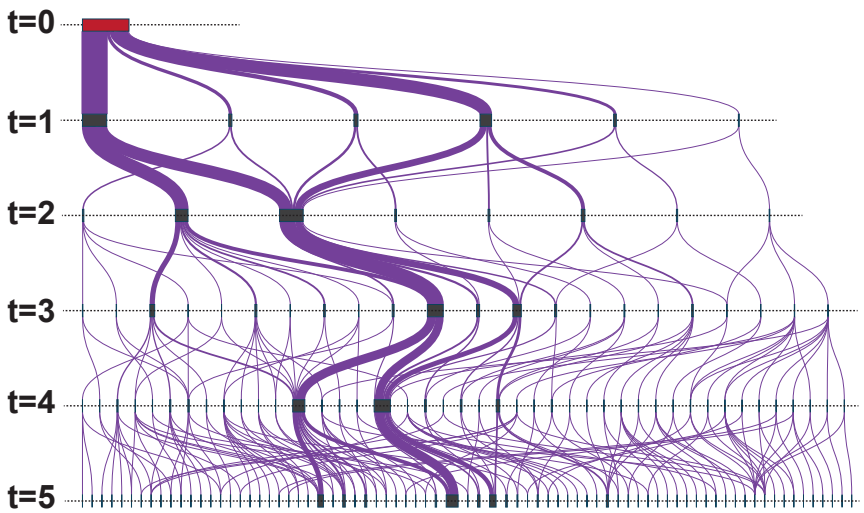
- The breaking of transitivity that is due to the arrow of time (and the chronological order in which time-stamped links occur) has important consequences. It implies that we cannot use most of the network analytic methods that we have used through the course anymore. Most of them are based on the (implicit or explicit) assumption that paths are transitive, i.e. that we can understand the topology of a system based on the **transitive hull of links** in the network.
- Note that it is this assumption that allows us to use the k-th power of a matrix to expand links to paths of length k. This is the reason why we can, for instance, use transition matrices to study random walks. From a different perspective, this is related to the Markovian property of the random walk process, where the next step of a random walk only depends on the current node.
- Time series data can exhibit temporal correlations in the chronological order of links that break this assumption of Markovianity, i.e. the next step on a causal path depends not only on the current node, but also on the previous node! Such higher-order dependencies can render commonly used network representations (and thus statistical network analysis and network-based machine learning algorithms) useless.
- From a statistical point of view, whenever we assume that the causal paths in a temporal network are a simple transitive closure of time-stamped edges, we assume that links are statistically independent (which is the same as assuming that the sequence of traversed nodes is Markovian).

Markovian vs. Non-Markovian paths



diffusion process

non-Markovian paths



diffusion process

Markovian paths

take-away message

non-Markovian paths can **slow down** or **speed up** diffusion process

→ I Scholtes et al., 2014

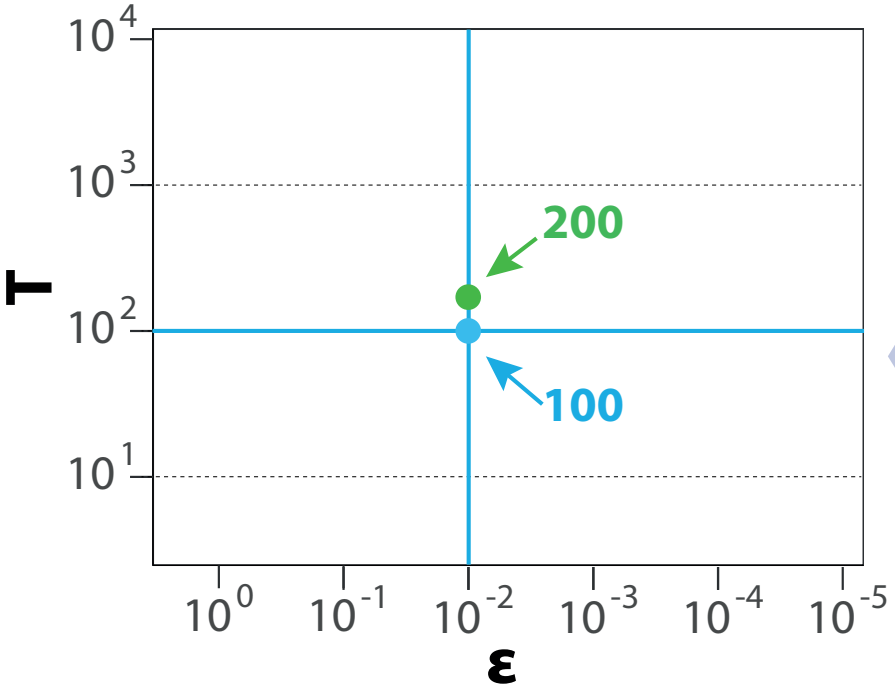
image credit: figures taken from R Lambiotte, M Rosvall, I Scholtes, 2019

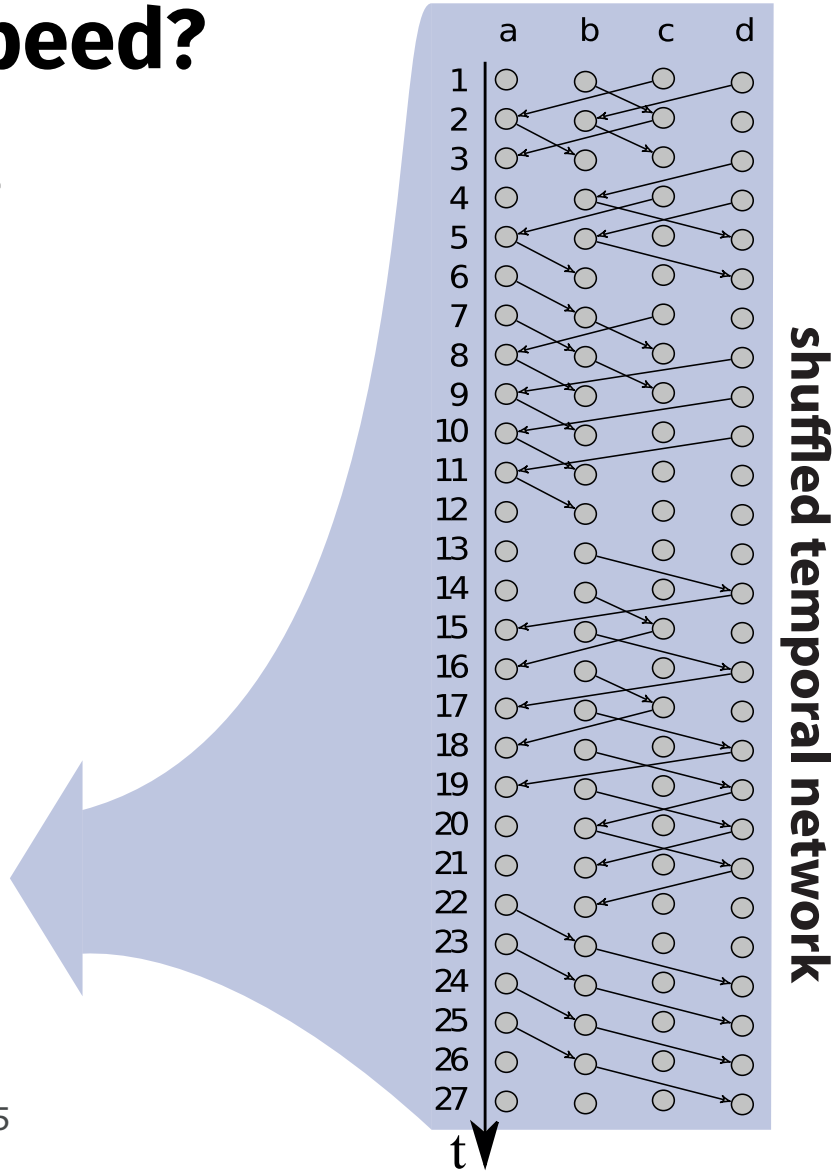
Notes:

- Such temporal correlations have important consequences for the evolution of dynamical processes, e.g. for the diffusion of conserved quantities. With a random walk model, we implicitly assume that all possible transitive paths exist (with a certain probability that depend on the multiplied transition probabilities). At the same time, we have seen that some of these transitive paths may not exist in a temporal network, due to the chronological ordering of time-stamped links.
- This explains why we often find diffusion processes (and other processes as well) to be slower in a temporal network compared to a time-aggregated network. In each step, the process simply has less options to proceed, due to the absence of some of the transitive paths.
- Remarkably, we also find examples where the opposite is the case, i.e. the diffusion process is actually faster in the temporal network (where only some of the transitive paths exist) compared to the time-aggregated network (where all transitive paths) exist.
- This intriguing finding shows that the effect of the temporal dynamics of networks on dynamic processes is not easy to understand (sometimes countering our intuition).
- The figure above shows the evolution of a diffusion process in the temporal network defined by dynamic train connections in the London Tube metro network. We show the first five steps of the process in a so-called alluvial diagram, where the width of the purple bar indicates the amount of conserved quantity (e.g. visitation probability) at a given node (horizontal ticks).

Influence on diffusion speed?

- how do non-Markovian paths influence **speed of diffusion process**?
- calculate convergence time T for total variation distance threshold ϵ

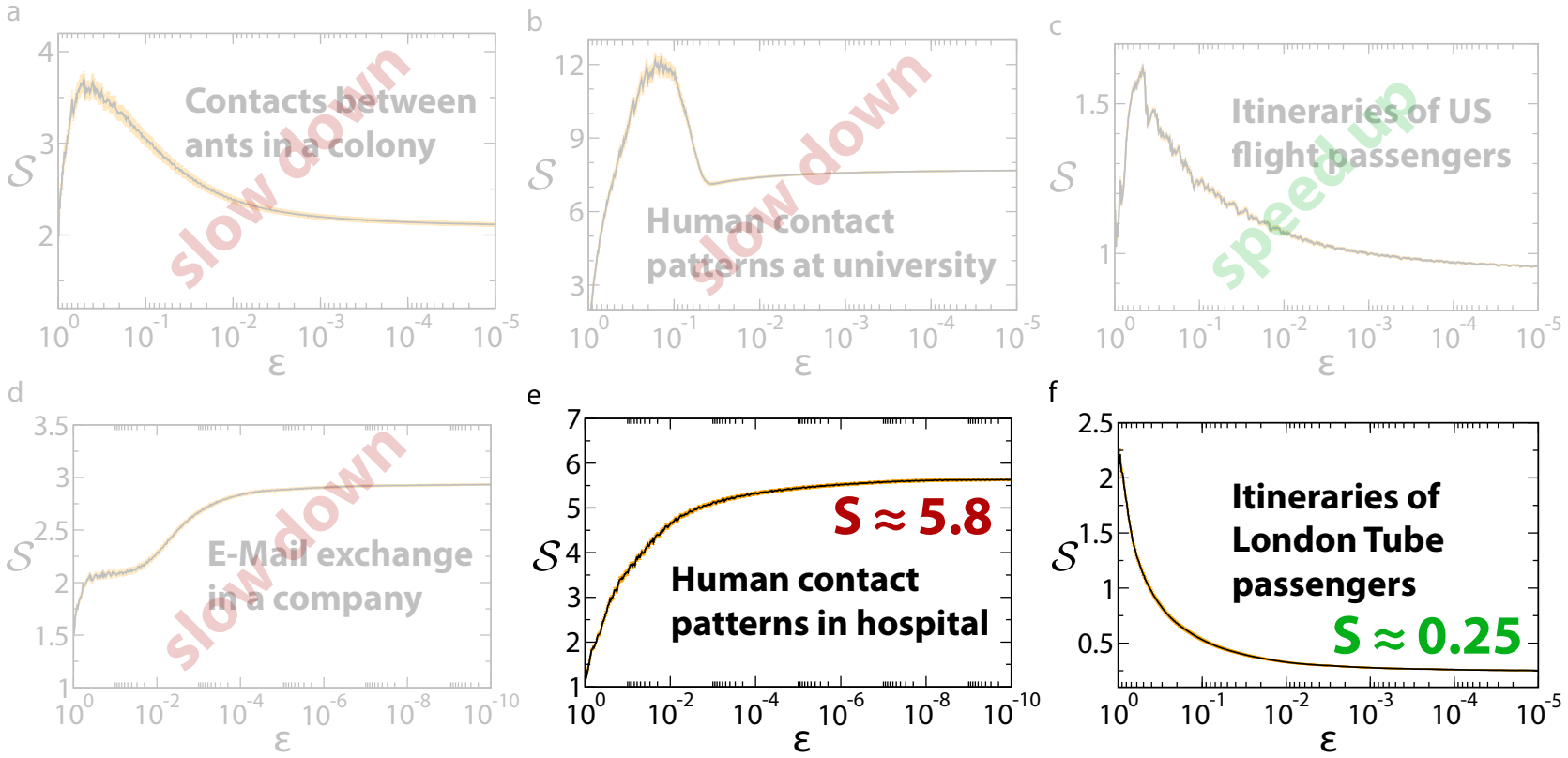




Notes:

- How can we quantify the effect of the chronological ordering of links in a temporal network on diffusion speed?
- For a static network, we have this by calculating the number of steps needed for random walk visition probabilities to converge to the stationary state (up to a given convergence threshold ϵ for the total variation distance to the stationary probabilities).
- For temporal networks, we can do the same. We can simulate a random walk on an empirical temporal network and calculate visitation probabilities after each step. This gives us the relaxation time T for a given threshold ϵ.
- To selectively study the effect of the chronological order of links, we can now randomly shuffle time stamps of links, and repeat the simulation, i.e. we calculate the relaxation time for a shuffled realisation of the model. We can now repeat this process many times, which allows us to calculate a slow-down factor S for each threshold ϵ, by dividing the value T in the empirical sequence by the value T in the shuffled sequence.
- Whenever S is smaller than one, the ordering of links speeds up diffusion (because the value T for the shuffled sequence is larger than for the empirical sequence). When S is larger than one, the ordering slows down the diffusion process.

Influence on dynamical processes?



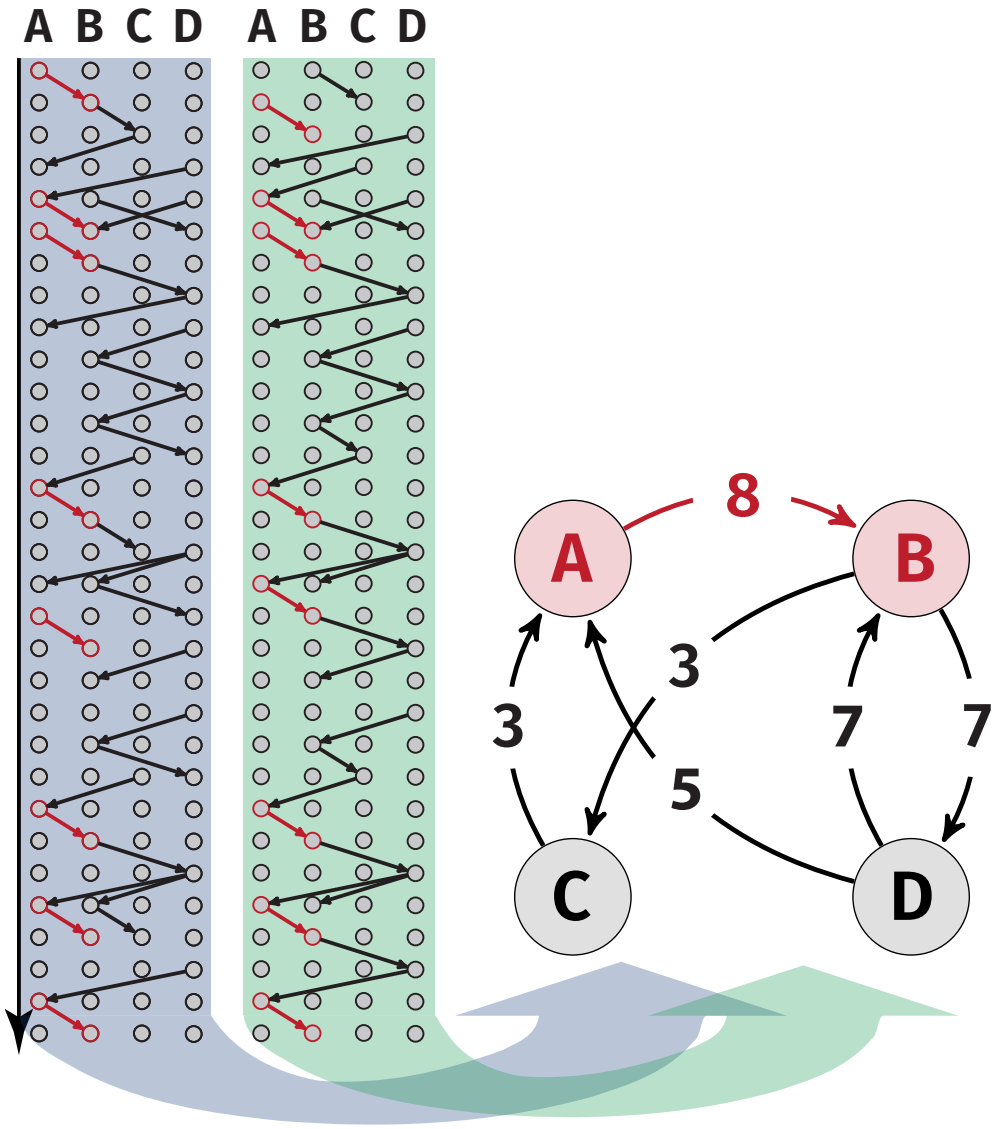
question

can we analytically explain different effects on diffusion speed?

Notes:

- Calculating the value for six different time series data (and using the value to which S converges for $\epsilon \to 0$), we find that the ordering of links has very different effects for different systems, both in terms of the direction of the change (i.e. speed up or slow down of diffusion) and in terms of the magnitude of S.
- This raises the question how we can analytically understand and predict those differences, which will hopefully also give us an approach to analyse temporal networks in a more meaningful way.

Network models for causal paths?



network model

max. entropy generative model for causal paths in a temporal network

projection of temporal network

network model is **projection of a directed acyclic graph** along time axis

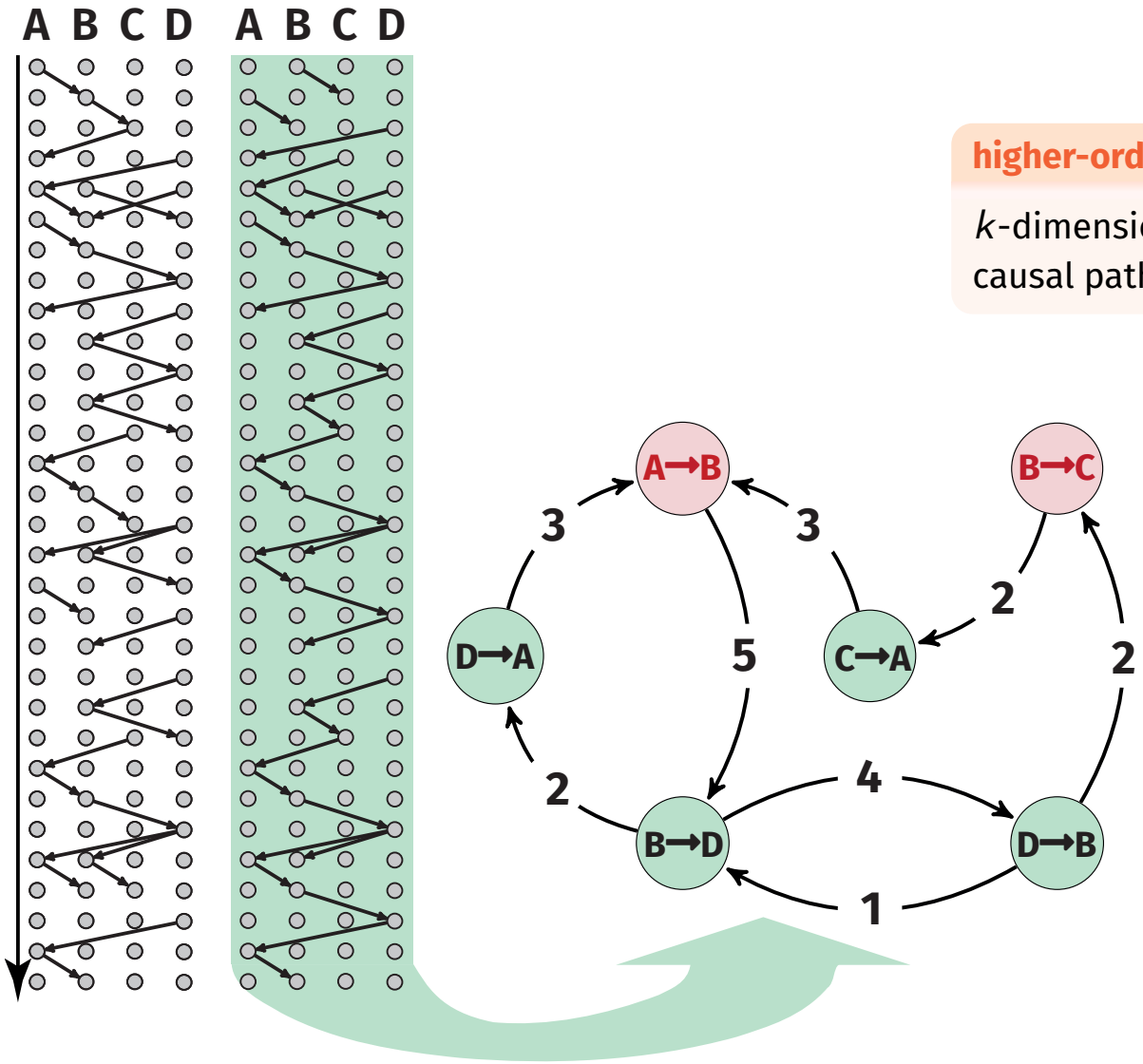
first-order network model

weights in time-aggregated network count **causal paths of length one**

Notes:

- A key problem with the study of time-aggregated networks is that we get a false sense of transitivity, since we discard all information on the ordering of links in the dynamic network, which influence causal paths. We should actually see time-aggregated networks as a maximum entropy model for causal paths in time series data, where we replace information on the actual causal paths in a **directed acyclic graph** by a maximally simple assumption of transitive and statistically independent links.
- We see this in the example above: both sequences give rise to the same time-aggregated network, even though links are ordered very differently, giving rise to two very different directed acyclic graphs.
- More formally: we cannot apply network-analytic or algebraic techniques to study time-stamped data, because the k-th power of the adjacency matrix of a time-aggregated network does not necessarily tell us something about the statistics of causal paths of length k in the directed acyclic graph above.
- Time-aggregated static networks are bad models for the structure (and statistics) of paths in temporal networks with order correlations.
- We can take a step back and view the commonly used time-aggregated networks in a different way: they can be thought of as static representations of time-stamped relational data that capture the statistics of time-stamped links, i.e. causal paths of length one. This is a first-order model of time series data, which only consider paths of length one.

Higher-order network models



higher-order network model

k-dimensional ***De Bruijn* graph model** of causal paths → I Scholtes et al., 2014

Notes:

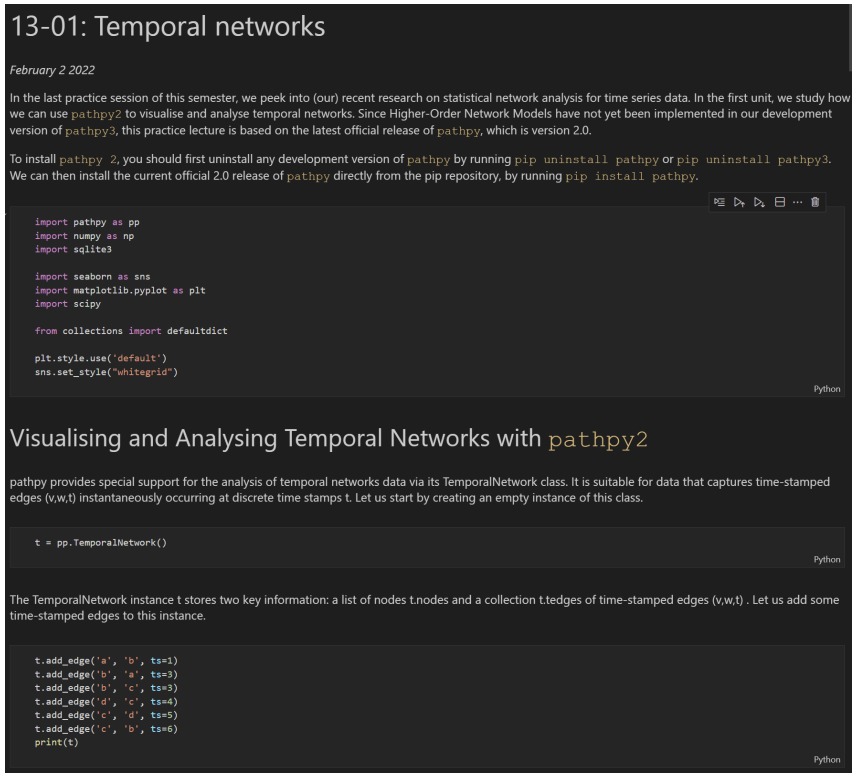
- The view of a time-aggregated network as first-order model that captures statistics of links (i.e. causal paths of length one) directly gives us an idea how we can extend this approach to study longer causal paths.
- We can turn the links of a first-order model into the nodes of a **second-order network model**. We can further connect two nodes in such a second-order model whenever the temporal network contains a causal path that consists of these two links. The weights of such links in a second-order model can then count the number of corresponding causal paths of length two.
- Such a graph model is actually a specific type of k-dimensional De Bruijn graphs, which have, e.g. been used to study genetic sequences. As a general rule, any two k-th order nodes in a k-th order network model must overlap in $k-1$ (first-order) nodes.
- We can see a first-order network model as a trivial example of a one-dimensional De Bruijn graph, where nodes do not overlap at all. The second-order model is the simplest possible higher-dimensional De Bruijn graph that captures the chronological ordering of links, i.e. if we change the ordering of links in the example above, the second-order network model changes while the first-order network model is the same.
- Clearly, a second-order network model cannot capture order correlations that influence the statistics of causal paths beyond a length of two, so we still lose some information on the actual causality in the temporal network. We can get around this problem by generalising our method to higher-order models (which capture the statistics of longer causal paths in the same way as a first- or a second-order model).

Practice Session

- we visualize **temporal networks** with `pathpy`
- we calculate **causal paths** in a temporal network
- we use **higher-order networks** to model those causal paths

practice session

see notebook 13-01 – 13-02 in `gitlab` repository at
→ `https://gitlab.informatik.uni-wuerzburg.de/ml4nets_notebooks/2021_wise_sna_notebooks`



13-01: Temporal networks

February 2 2022

In the last practice session of this semester, we peek into (our) recent research on statistical network analysis for time series data. In the first unit, we study how we can use `pathpy2` to visualise and analyse temporal networks. Since Higher-Order Network Models have not yet been implemented in our development version of `pathpy3`, this practice lecture is based on the latest official release of `pathpy`, which is version 2.0.

To install `pathpy 2`, you should first uninstall any development version of `pathpy` by running `pip uninstall pathpy` or `pip uninstall pathpy3`. We can then install the current official 2.0 release of `pathpy` directly from the pip repository, by running `pip install pathpy`.

```python
import pathpy as pp
import numpy as np
import sqlite3

import seaborn as sns
import matplotlib.pyplot as plt
import scipy

from collections import defaultdict

plt.style.use('default')
sns.set_style("whitegrid")
```

Python

Visualising and Analysing Temporal Networks with `pathpy2`

pathpy provides special support for the analysis of temporal networks data via its TemporalNetwork class. It is suitable for data that captures time-stamped edges (v,w,t) instantaneously occurring at discrete time stamps t. Let us start by creating an empty instance of this class.

```python
t = pp.TemporalNetwork()
```

Python

The TemporalNetwork instance t stores two key information: a list of nodes t.nodes and a collection t.tedges of time-stamped edges (v,w,t) . Let us add some time-stamped edges to this instance.

```python
t.add_edge('a', 'b', ts=1)
t.add_edge('b', 'a', ts=3)
t.add_edge('b', 'c', ts=3)
t.add_edge('d', 'c', ts=4)
t.add_edge('c', 'd', ts=5)
t.add_edge('c', 'b', ts=6)
print(t)
```

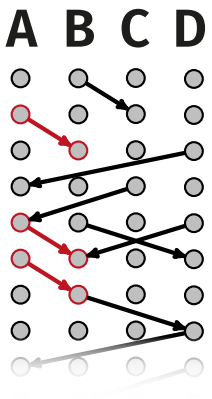
Python

Notes:

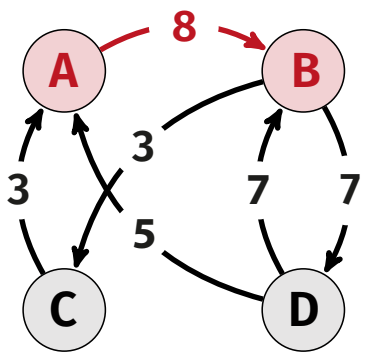
- In the first practice session, we show how we can use pathpy to create and visualize networks and to calculate causal paths. We further show how we can create higher-order models with different orders to model such paths.

Higher-order spectral analysis

temporal network



first-order topology



second-order topology

$$\mathbf{T} = \begin{array}{c} \\ a;b \\ b;c \\ d;a \\ c;a \\ b;d \\ d;b \end{array} \begin{array}{c} \begin{array}{cccccc} a;b & b;c & d;a & c;a & b;d & d;b \end{array} \\ \begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Markovian second-order topology

$$\tilde{\mathbf{T}} = \begin{array}{c} \\ a;b \\ b;c \\ d;a \\ c;a \\ b;d \\ d;b \end{array} \begin{array}{c} \begin{array}{cccccc} a;b & b;c & d;a & c;a & b;d & d;b \end{array} \\ \begin{bmatrix} 0 & \frac{1}{3} & 0 & 0 & \frac{2}{3} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & 0 & 0 & \frac{2}{3} & 0 \end{bmatrix} \end{array}$$

$\lambda_2(\mathrm{T})$ ← **diffusion speed** → $\lambda_2(\tilde{\mathrm{T}})$

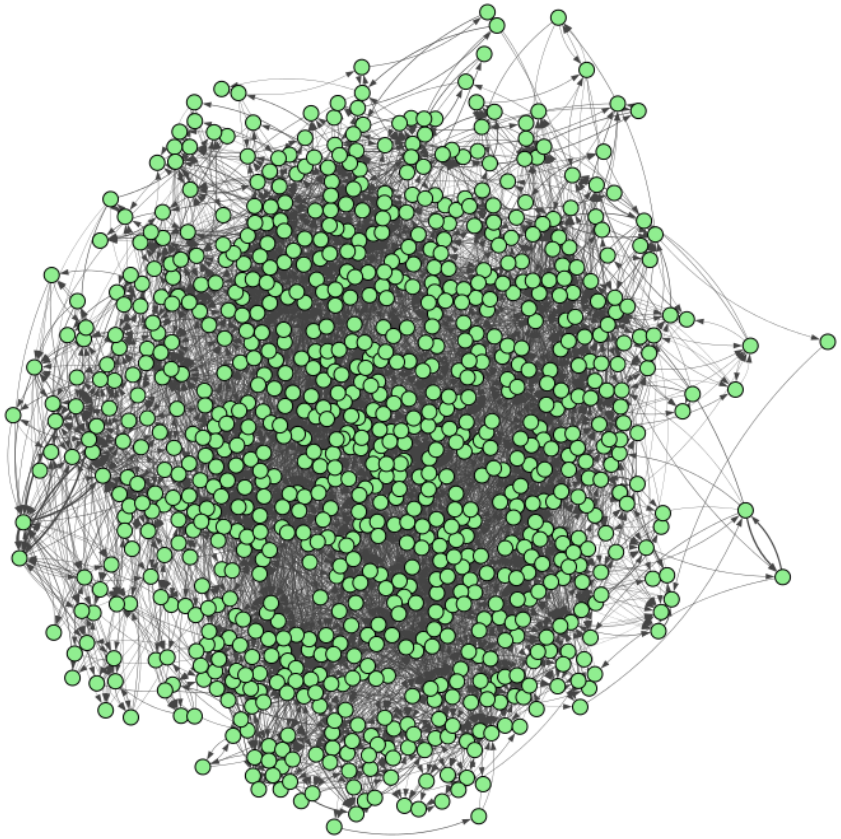
Notes:

- An interesting aspect of higher-order network models is that they provide a method to generalize the analytical study of diffusion processes to temporal networks. Remember that in lecture L10, we interpreted a random walk on a (static) graph as a stochastic model for paths in a network. Following the same idea we can interpret a random walk on a higher-order network as **stochastic model for causal paths in a temporal network**. This model accounts for higher-order dependencies (i.e. it does not necessarily assume transitivity in the first-order topology) and is thus a more accurate representation of causal paths.

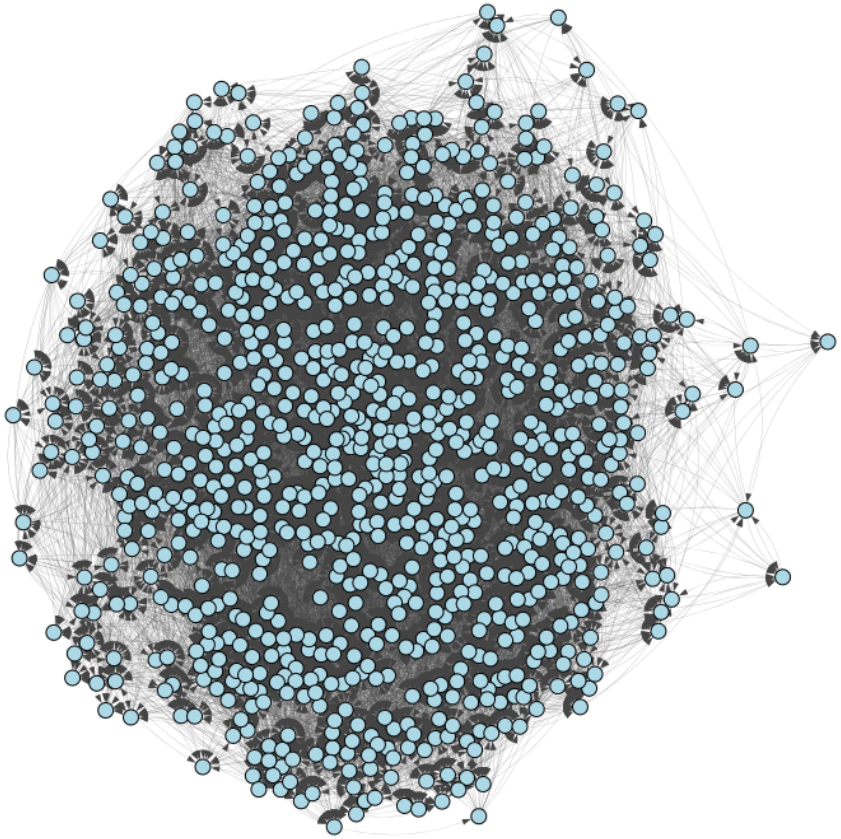
- For a given time-stamped data set we can further easily calculate *expected* higher-order topologies, i.e. the statistics and topology of causal paths which we implicitly assume based on a time-aggregated perspective. This allows us to compare the actual second-order topology in the time-stamped data to the one expected based on the first-order topology. This difference captures how the temporal ordering of links changes path structures in a temporal network, compared to the time-aggregated network!

- We can then use spectral methods to quantify the connectedness of the topologies as shown in previous lectures, i.e. we can calculate the second-largest eigenvalue $|\lambda_2(\mathbf{T})|$ of the transition matrix of a (biased) random walk process. The eigenvalue gap $1 - |\lambda_2(\mathbf{T})|$ of a second-order transition matrix captures how "well-connected" the causal topology of the temporal network is. We can now compare the eigenvalue gap of the real second-order network to the eigenvalue gap of the "expected" second-order network, i.e. a model where all transitive paths exist.

Spectral gap in dynamic social network



real second-order topology



Markovian second-order topology

data: dynamic contact patterns between hospital workers

Notes:

- In lecture 12, we found that the larger the eigenvalue gap $1 - |\lambda_2(\mathbf{T})|$, the "better connected" the network and the faster a diffusion process will evolve.
- In the figure above, we show the second-order network model of causal paths in the temporal network representing dynamic contact patterns between hospital workers. The plot on the left shows the actual second-order network, i.e. here nodes are actually links in the underlying network. We find that the second-largest eigenvalue is approx. 0.957.
- We can compare this to the plot of the expected second-order network model of Markovian causal paths, i.e. here we have assumed that each causal path of length two (which is represented by a link in the second-order network) appears with the probability given by a Markov chain model. In other words, we can factorize the probability $P(c|a \to b) = P(c|b) \cdot P(b|a)$, i.e. the links are occurring independently from each other. The resulting second-largest eigenvalue is larger than that of the real network, which captures the non-surprising fact that the causal topology of a dynamic network in which all transitive causal paths that can possibly occur actually occur is more connected than a dynamic network in which this is not the case.
- How can we use this to analytically predict how this change in the causal topology affects diffusion speed?

Predicting the slow-down factor

- we can study random walks in networks in a "higher-order state space"

  $\longrightarrow$ exercise sheet 09

- in Lecture 12, for the diffusion speed of a random walk process we found:

$$\delta(\pi, \pi^{(t)}) \leftrightarrow t \geq \frac{\log(\epsilon)}{\log(|\lambda_2(T)|)}$$

- for the **slow-down factor** that is due to causal paths in a temporal network compared to paths in a static network we thus predict

$$S = \frac{\frac{\log(\epsilon)}{\log(|\lambda_2(T)|)}}{\frac{\log(\epsilon)}{\log(|\lambda_2(\tilde{T})|)}} = \frac{\log(|\lambda_2(\tilde{\mathbf{T}})|)}{\log(|\lambda_2(\mathbf{T})|)}$$

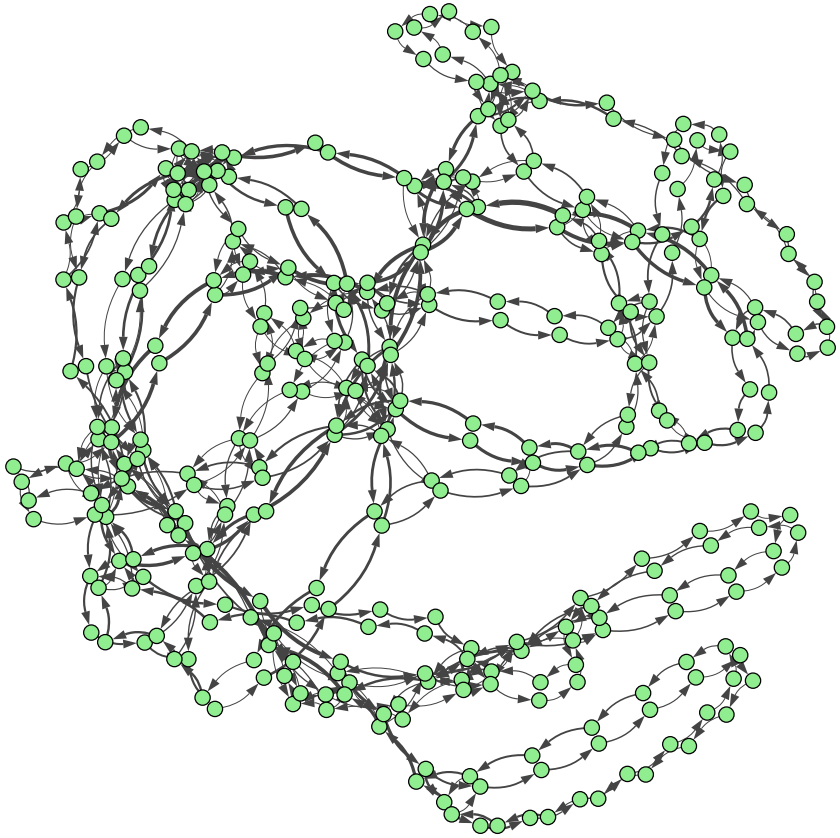
- for the social network from the previous slide, we find

$$S = \frac{\log(|\lambda_2(\tilde{\mathbf{T}})|)}{\log(|\lambda_2(\mathbf{T})|)} = 5.75$$

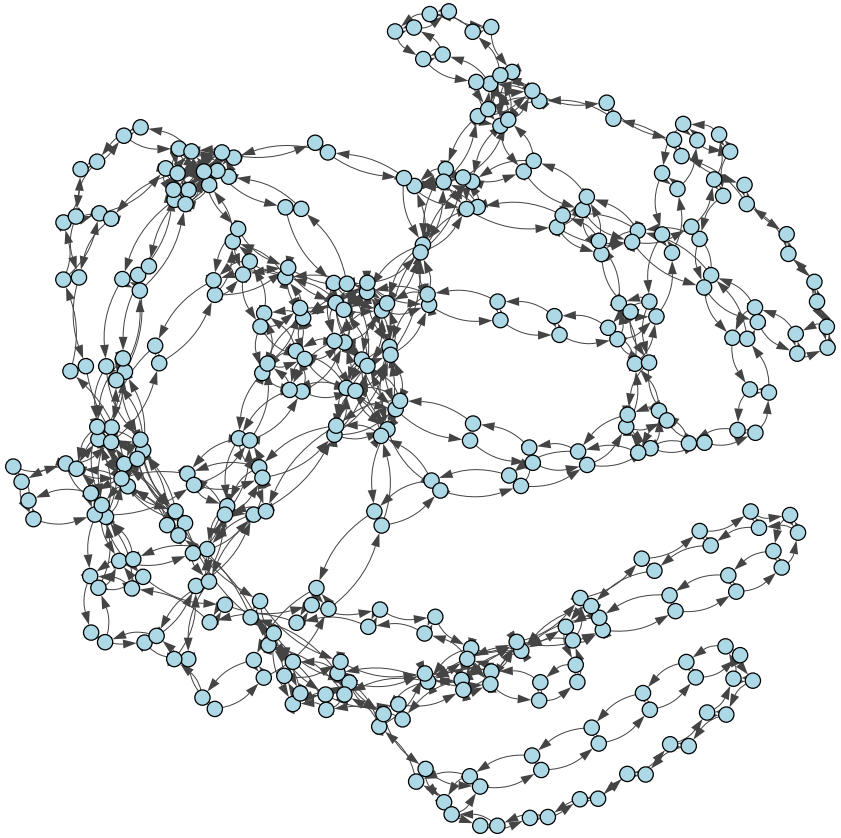
Notes:

- We can analytically show that the ratio between the speed of diffusion in two network topologies can be calculated based on the ratio between the logarithms of the two corresponding eigenvalues.
- For this, we remember that the minimum number of random walk steps until visitation probabilities reach the stationary state up to a total variation distance smaller than ϵ is proportional to the log of epsilon divided by the log of the absolute value of the second-largest eigenvalue of the transition matrix $\longrightarrow$ Lecture 12
- We can use this calculate an analytical prediction of a slow-down factor, which is defined as the ratio of the diffusion time in the actual second-order network model of causal ppaths in a temporal network compared to the diffusion time in the expected second-order network model, where all transitive paths exist. This yields a simple prediction based on the ratio of the logs of the two eigenvalues.
- For the example of the dynamic social network in the hospital, we obtain a ratio of 5.75. This ratio captures the expected slow down of diffusion that is due to the ordering of time-stamped links in the data set (and the resulting effects on the causal topology). This prediction nicely corresponds to the actual slow dow factor of $S = 5.8$, which we have calculated based on our simulations of a diffusion process in the temporal network (see slide 9).
- For details see the paper:
- I Scholtes et al.: “Causality-driven slow-down and speed-up of diffusion in

London Tube: speeding up diffusion



real second-order topology



Markovian second-order topology

$$S = \frac{\ln|\lambda_2(\tilde{\mathbf{T}})|}{\ln|\lambda_2(\mathbf{T})|} = 0.24$$

Notes:

- If we apply the same approach to the data on dynamic train connections in the London Tube network, we analytically derive a slow-down factor of $S = 0.24$, which again corresponds to the speed up by a factor of four which we have observed in our simulations (slide 9).
- This simple approach works in a large variety of time-stamped data sets covering different types of networked systems with dynamic topologies, i.e. we can use it to analytically predict the speed of a diffusion process in a dyanmic network topology.
- Moreover, we can use it to quantitatively study how “well-connected” the causal topology (i.e. the topology of nodes connected by causal paths) in a temporal network actually is, and how this causal topology differs from the time-aggregated network.

Time-aware centrality measures

centrality measures

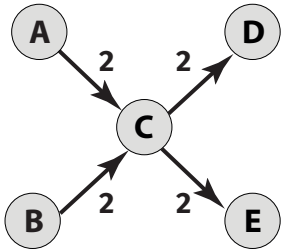
- find "influential" individuals
  - **spectral centralities**
  - **path-based centralities**
- but: focus on **topological** importance

temporal-topological centrality

- ordering of links changes **temporal influence** of nodes → H Habiba et al., 2007
- **higher-order centralities** capture influential nodes in temporal networks
  → I Scholtes, N Wider, A Garas, 2016

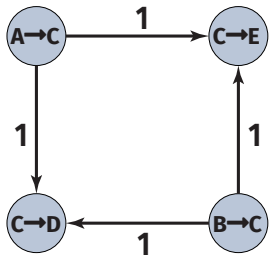
example

time-stamped interactions between software developers in OpenSource project Eclipse



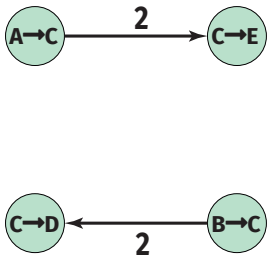
first-order node centralities

betweenness(**C**) = 4



second-order model 1

betweenness(**C**) = 4



second-order model 2

betweenness(**C**) = 2

Notes:

- An interesting further application of higher-order network models are time-aware measures for the centrality of nodes in a temporal network.
- We can generalize standard (static) node centrality measures to higher-order networks, e.g. we calculate these centralities in the (static) second-order network. We obtain centrality scores of **second-order nodes**, which correspond to links in the network. We can then project them to node centralities and obtain new centrality measures that capture both the topology of links, and the ordering in which these links appear. For details, please refer to → I Scholtes et al., 2016
- To validate the resulting measure, we can calculate centralities of nodes in a temporal network, based on an analysis of all causal paths (this is computationally expensive and rather cumbersome, however it provides us with "ground truth" centrality scores of nodes).
- We can then check how well the centrality score of nodes in higher-order networks with different orders correspond to the true node temporal node centralities. We find that higher-order centralities with orders larger than one better capture the importance of nodes in temporal networks (see details in paper referenced above).

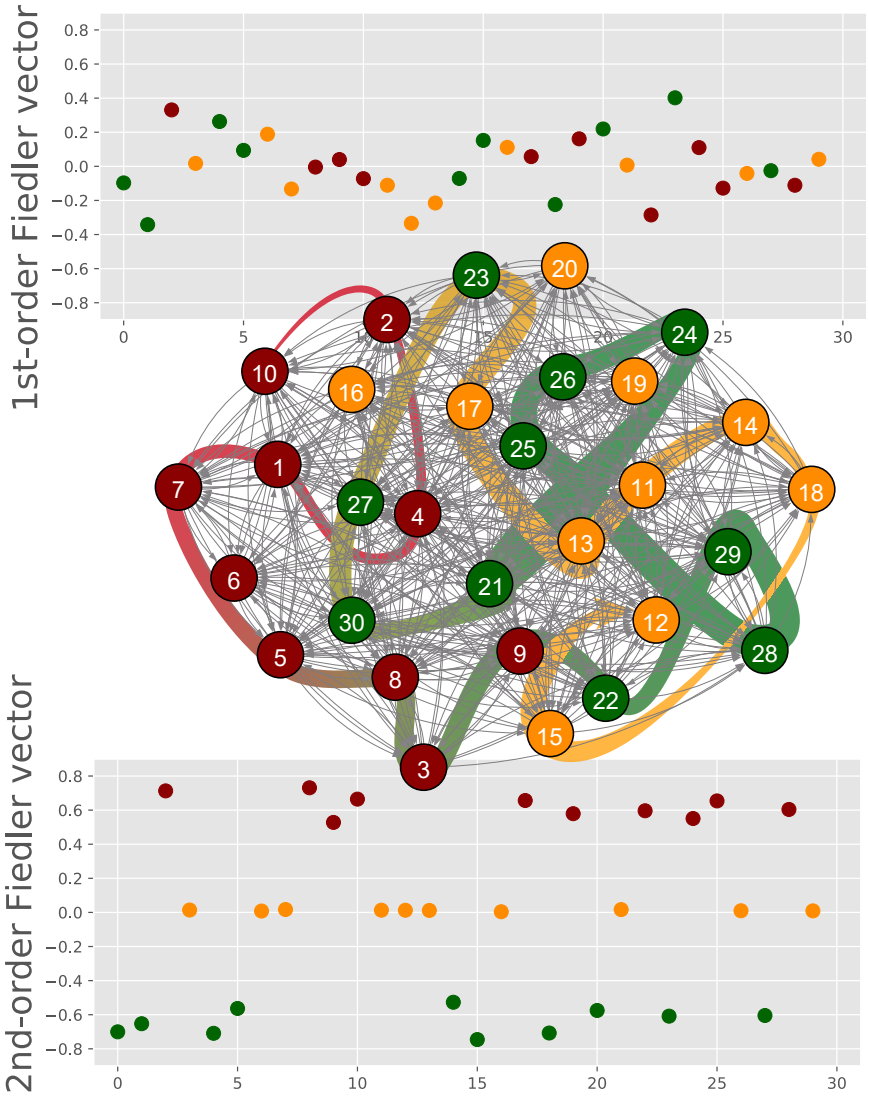
Time-aware spectral clustering

graph clustering algorithms

- find clusters of "well-connected" nodes
  - **modularity optimisation**
  - **clique percolation**
  - **flow compression**
  - **spectral clustering**
- but: focus on **topological** communities

temporal-topological clusters

- ordering of links can introduce **clusters in causal topology**
- spectral clustering in **higher-order Laplacians**
  → I Scholtes et al., 2014
  → practice lecture P12
- **higher-order flow compression**
  → M Rosvall et al. 2014



Notes:

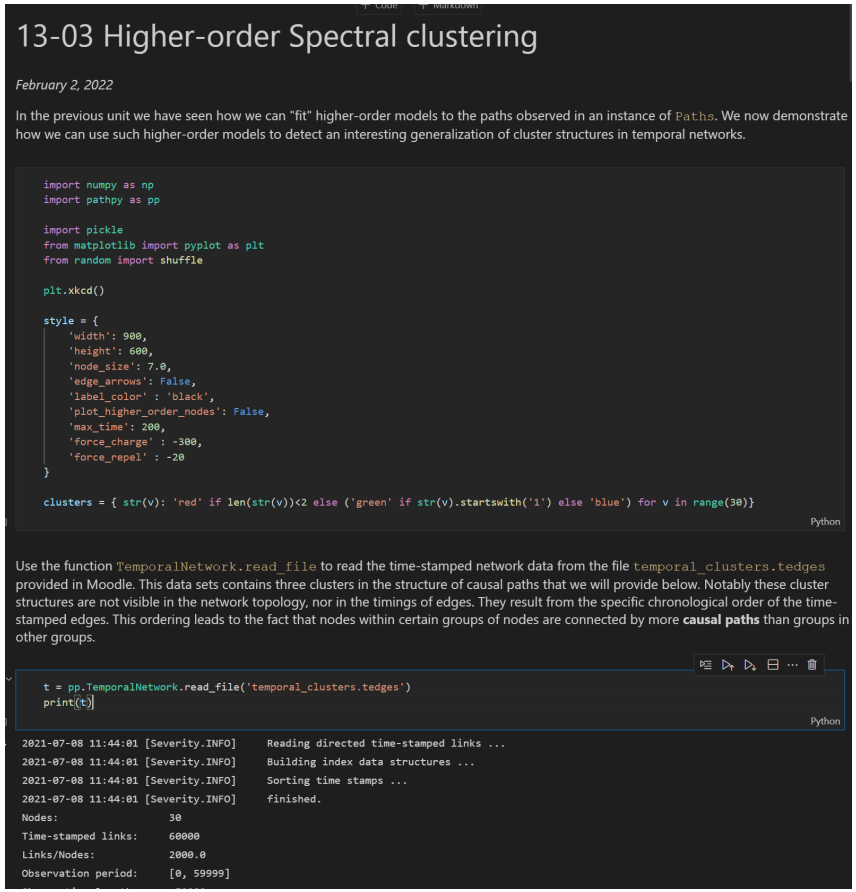
- As another application of higher-order network we consider **time-aware graph clustering algorithms** for time series data on networks.
- As mentioned in → lecture L12 we can view natural communities as groups of nodes within which a random walker is "trapped" for extended periods of time. It turns out that this trapping cannot only be due to the topology, but also due to the ordering of time-stamped links.
- Applying our approach of higher-order models, we can study the spectrum of matrix representations of higher-order topologies, in particular we can detect temporal-topological clusters by **clustering entries of a higher-order Fiedler vector**, i.e. the eigenvector v_2 corresponding to the second-smallest eigenvalue of a Laplacian of a higher-order network.
- This approach provides us with an interesting new definition of clusters in time-stamped relational data, which can be applied in biological networks, information systems, and social networks.

Practice Session

- we study a synthetic example for a **temporal network with a clustered causal topology**
- we use the **spectrum of a higher-order Laplacian** to detect those clusters
- we show that cluster patterns can emerge only due to the **temporal ordering of links**

practice session

see notebook 13-03 in `gitlab` repository at
→ `https://gitlab.informatik.uni-wuerzburg.de/ml4nets_notebooks/2022_wise_sna_notebooks`



13-03 Higher-order Spectral clustering

February 2, 2022

In the previous unit we have seen how we can "fit" higher-order models to the paths observed in an instance of `Paths`. We now demonstrate how we can use such higher-order models to detect an interesting generalization of cluster structures in temporal networks.

```python
import numpy as np
import pathpy as pp

import pickle
from matplotlib import pyplot as plt
from random import shuffle

plt.xkcd()

style = {
    'width': 900,
    'height': 600,
    'node_size': 7.0,
    'edge_arrows': False,
    'label_color' : 'black',
    'plot_higher_order_nodes': False,
    'max_time': 200,
    'force_charge' : -300,
    'force_repel' : -20
}

clusters = { str(v): 'red' if len(str(v))<2 else ('green' if str(v).startswith('1') else 'blue') for v in range(30)}
```

Python

Use the function `TemporalNetwork.read_file` to read the time-stamped network data from the file `temporal_clusters.tedges` provided in Moodle. This data sets contains three clusters in the structure of causal paths that we will provide below. Notably these cluster structures are not visible in the network topology, nor in the timings of edges. They result from the specific chronological order of the time-stamped edges. This ordering leads to the fact that nodes within certain groups of nodes are connected by more **causal paths** than groups in other groups.

```python
t = pp.TemporalNetwork.read_file('temporal_clusters.tedges')
print(t)
```

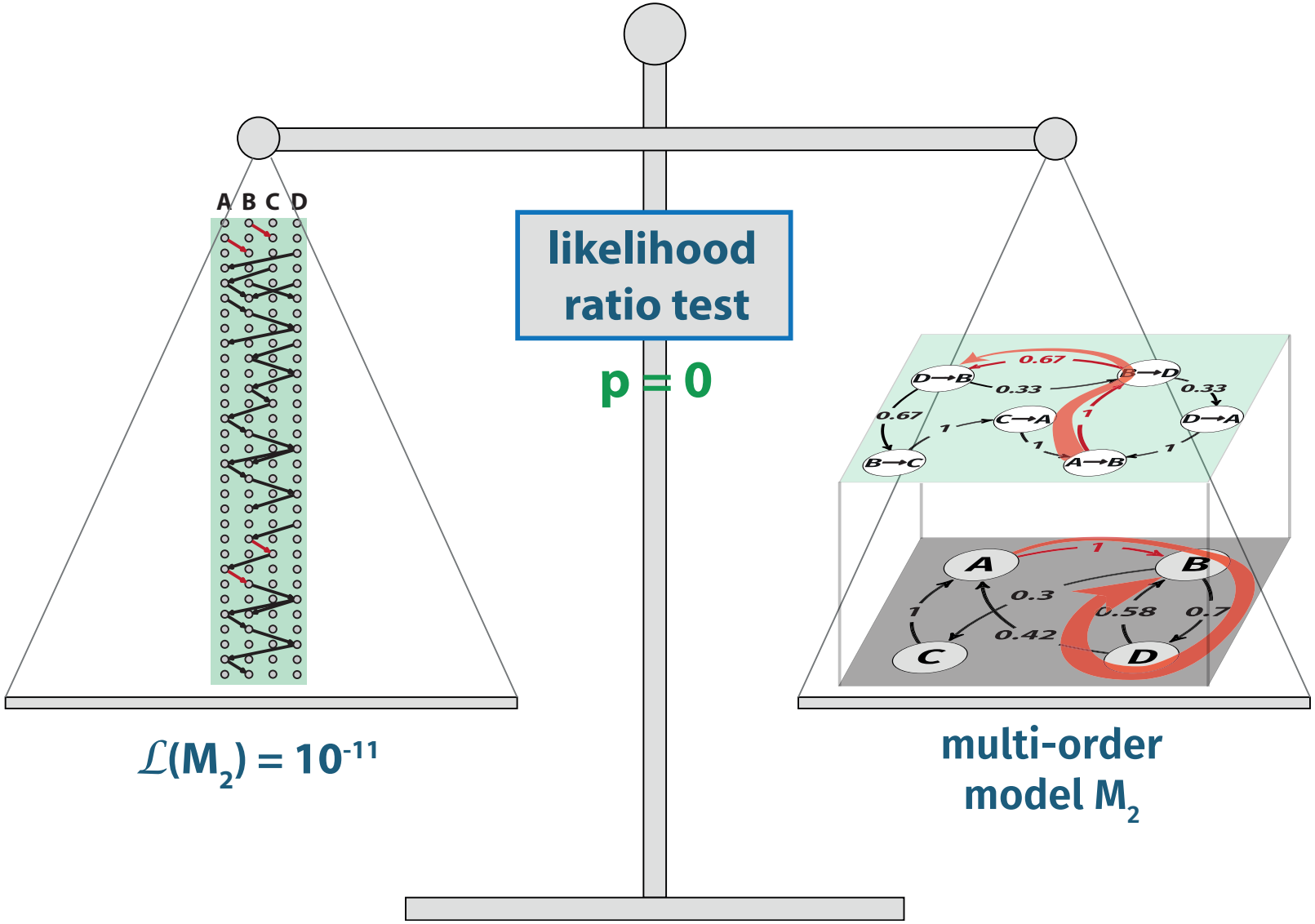
Python

```
2021-07-08 11:44:01 [Severity.INFO]     Reading directed time-stamped links ...
2021-07-08 11:44:01 [Severity.INFO]     Building index data structures ...
2021-07-08 11:44:01 [Severity.INFO]     Sorting time stamps ...
2021-07-08 11:44:01 [Severity.INFO]     finished.
Nodes:                  30
Time-stamped links:     60000
Links/Nodes:            2000.0
Observation period:     [0, 59999]
```

Notes:

- In the second practice session, we show how we can use spectral methods to detect clusters in the causal topology of temporal networks. We specifically show that such clusters can be destroyed by a reshuffling of time stamps, which proves that those clusters are not due to the topology or frequency of interactions.

Model selection?



$\mathcal{L}(M_2) = 10^{-11}$

multi-order model M_2

→ Machine Learning for Networks

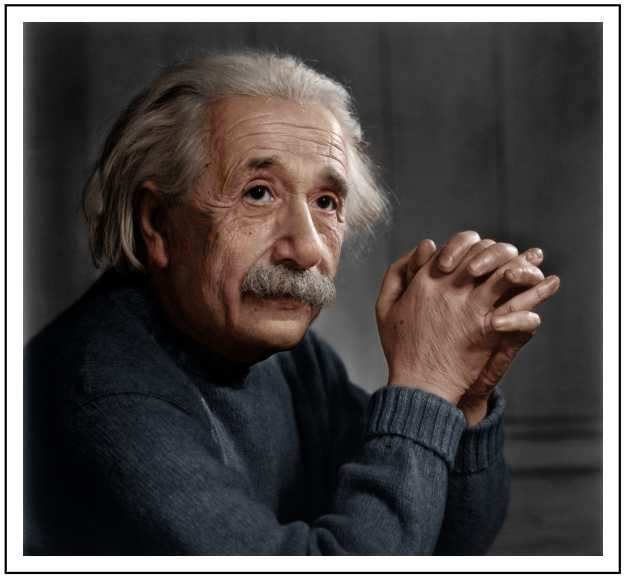
Notes:

- In the previous examples, we have shown that we can use second-order network models to understand dynamical processes, define new time-aware measures of centrality, or detect cluster patterns in temporal networks. However, this also raises two crucial questions that we have not addressed, namely which order of a network model we should use to analyse a given time series data set. Specifically, we want to know:

    1. When is a standard (first-order) network model sufficient to study time series data?
    2. What is the optimal order to study a given time series data set?

- If we are able to answer the second question, we automatically also solve the first one (which reduces to the question whether the optimal order of a model is one!). We can treat this as a model selection problem, i.e. similar to other machine learning tasks we need to find the optimal model that best generalizes to unseen data.
- This can be addresased by treating higher-order models as stochastic, Markov chain models of paths, which allows us to calculate a likelihood of a model given a set of observed causal paths. We can then apply a likelihood ratio test that accounts for the complexity of the models. The following cartoon video explains the method and the underlying motivation: `https://www.youtube.com/watch?v=CxJkVrD2ZlM`
- The details of this method are beyond the scope of this course. If you want to learn more about model selection and the likelihood ratio test, I must refer you to introductory courses on data science or to our upcoming course → Machine Learning for Networks in the summer semester. Details can be found in → I Scholtes, 2017

Learning network models from data



"Make things as simple as possible, but not simpler."

Albert Einstein

"Pluralitas non est ponenda sine necessitate."

William of Ockham

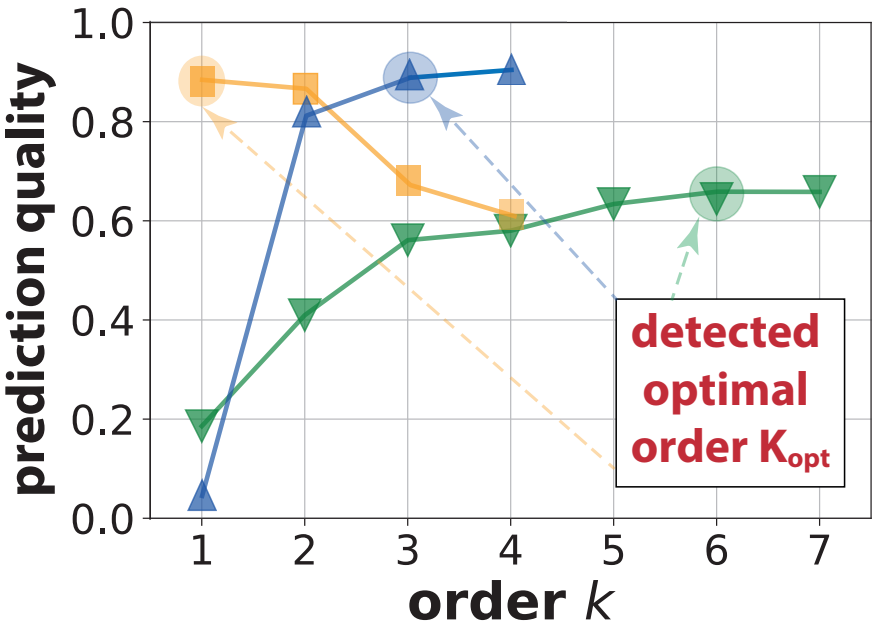
take-away message

need methods to learn **optimal order** of a network model → I Scholtes, 2017

Notes:

- Here it is enough to say that the likelihood ratio test is a practical method to operationalise Occam's razor, i.e. it allows us to find a maximally simple stochastic model that still explain the causal paths that we observe in time series data.
- From this point of view, a first-order, time-aggregated model is too simple to explain the complex causal paths that we see in real time series data. A first-order model thus underfits such time series data. On the other hand, we could always find an overfitting model that trivially encodes the causal paths in a data set in a higher-order model with very high order.
- The method proposed in the paper mentioned above balances between under- and overfitting. It gives a method to learn optimal higher-order models from time series data.
- With this slide, we also close a circle to the very first lecture, in which I gave an overview of the scientific method. A key challenge in science is to find parsimonious models/theories/explanations for phenomena that make as few assumptions as possible and that generalize beyond the specific phenomenon that they seek to explain. In other words, we are interested in models that we can use to make predictions.

Optimal models!



- time-stamped network data from **technical and social systems**
- how well do centralities in "learned" model predict ground truth node importance?

conclusion

models with detected **optimal order yield best prediction** → I Scholtes, 2017

Notes:

- Let me finally show that the order detected by the approach summarised above actually corresponds to the “optimal” order needed to study a given data set. We can do this by means of a cross-validation, where we learn a model from a training data set and then test how well it predicts a “ground truth” in a validation data set.
- We can actually measure the ground truth importance of nodes in a validation data set (by generalising path-based centralities to causal paths in a temporal network). We then create different higher-order models based on a training data set, and calculate path-based centralities of nodes in these models. We can now calculate how well the centralities of nodes in the higher-order models allow us to predict the ground truth centralities of nodes.
- We find that first-order models are often very bad models for the true centrality of nodes, e.g. in social networks or transportation networks. We also find that the predictive power of higher-order models is maximised exactly at the order detected by our model selection technique. In other words, models with the detected order generalise best to a validation data set. For details of this analysis see the paper mentioned above.

Practice Session

- we calculate the **likelihood of higher-order models** for a given set of causal paths
- we use hypothesis testing to **select the optimal order of a model**
- we apply model selection to **optimal-order spectral clustering**

practice session

see notebooks 13-04 – 13-05 in `gitlab` repository at

→ `https://gitlab.informatik.uni-wuerzburg.de/ml4nets_notebooks/2022_wise_sna_notebooks`

13-04 Multi-Order Model Selection

February 2, 2022

We can interpret different higher-order models as **probabilistic generative models** for the paths that we have observed. Conveniently, the `HigherOrderNetwork` class in `pathpy` comes with a method `likelihood`. It calculates the probability to observe exactly the paths stored in a given `Paths` instance, given a higher-order network model. This allows us to calculate the likelihood of the higher-order model given the observed paths (cf. L10 on statistical inference).

We have considered higher-order network models for paths with a fixed order k. But how can we decide which order we should use to model a given data set? And how do we decide whether path statistics deviates significantly from the transitive, Markovian assumption made by a first-order model in the first place. We thus need methods to decide when higher-order models are actually needed and which is the optimal order to model paths.

Moreover, a higher-order model with order k can only capture higher-order dependencies at a single fixed correlation length k . But we may encounter data that exhibit multiple correlation lengths at once. How can we combine models with multiple higher orders into a multi-order model?

In this unit, we take a statistical inference and machine learning perspective to answer these questions. To show how the method works, we again start with a maximally simple toy example:

```python
import pathpy as pp
```

Python

```python
paths_1 = pp.Paths()
paths_1.add_path(('a','c','d'), frequency=2)
paths_1.add_path(('b','c','e'), frequency=2)

n = pp.Network.from_paths(paths_1)
n
```

Python

In the example `paths_1` we only observe two of the four paths of length two that would be possible in the null model. Hence, this is an example for path statistics that exhibit correlations that warrant a second-order model.

But how can we decide this in a meaningful way? We can take a statistical inference perspective on the problem. More specifically, we will consider our higher-order networks as probabilistic generative models for paths in a given network topology. For this, let us use the weighted first-order network model to construct a transition matrix of a Markov chain model for paths in a network. We simply use the relative frequencies of edges to proportionally scale the probabilities of edge transitions in the model.

```python
hon_1=pp.HigherOrderNetwork(paths_1, k=1)
pp.visualisation.plot(hon_1)
print(hon_1.transition_matrix().todense())
```

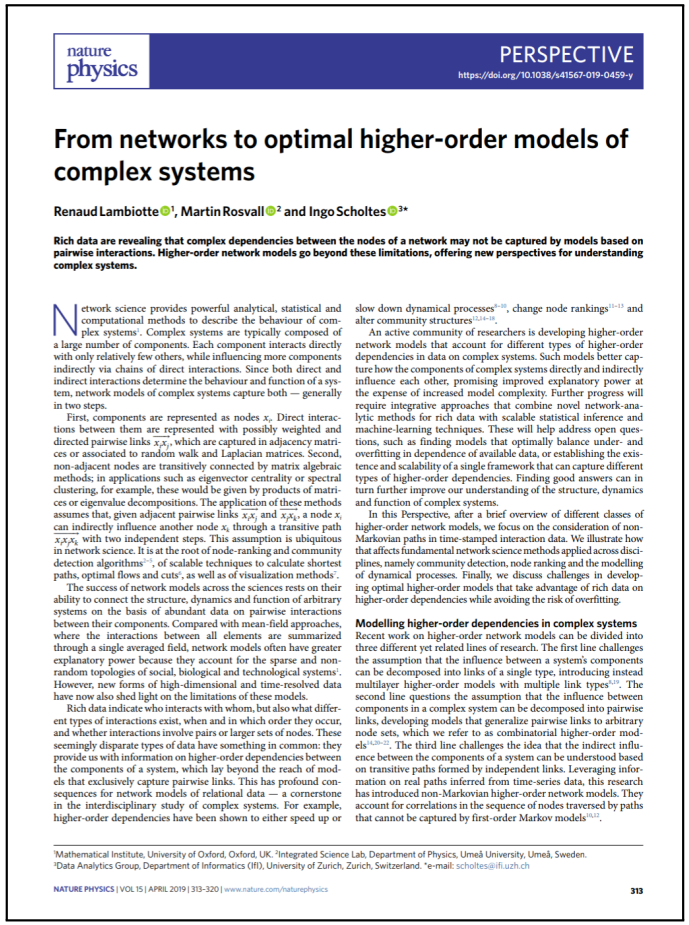
Python

Notes:

- In the last practice session of this lecture and - in fact - of our course, I demonstrate how we can caclulate likelihoods of higher-order network models, and how we can use the likelihood ratio test to detect the optimal maximum order of a multi-order network model. Linking this method with the previous practice session, we also show that we can use it to detect whether a temporal network contains temporal correlations that warrant a second-order model, e.g. for clustering or visualisation.

From networks to higher-order models

- network models capture **topology of dyadic interactions**
- arrow of time influences **causal topology**
- **higher-order network models** encode both time and topology
- `pathpy` supports **higher-order model selection and analysis**
- see Nature Physics perspective for more applications of higher-order network models



nature physics

PERSPECTIVE

https://doi.org/10.1038/s41567-019-0459-y

From networks to optimal higher-order models of complex systems

Renaud Lambiotte[1], Martin Rosvall[2] and Ingo Scholtes[3]*

Rich data are revealing that complex dependencies between the nodes of a network may not be captured by models based on pairwise interactions. Higher-order network models go beyond these limitations, offering new perspectives for understanding complex systems.

Network science provides powerful analytical, statistical and computational methods to describe the behaviour of complex systems[1]. Complex systems are typically composed of a large number of components. Each component interacts directly with only relatively few others, while influencing more components indirectly via chains of direct interactions. Since both direct and indirect interactions determine the behaviour and function of a system, network models of complex systems capture both — generally in two steps.

First, components are represented as nodes x_i. Direct interactions between them are represented with possibly weighted and directed pairwise links $\overrightarrow{x_i x_j}$, which are captured in adjacency matrices or associated to random walk and Laplacian matrices. Second, non-adjacent nodes are transitively connected by matrix algebraic methods; in applications such as eigenvector centrality or spectral clustering, for example, these would be given by products of matrices or eigenvalue decompositions. The application of these methods assumes that, given adjacent pairwise links $\overrightarrow{x_i x_j}$ and $\overrightarrow{x_j x_k}$, a node x_i can indirectly influence another node x_k through a transitive path $\overrightarrow{x_i x_j x_k}$ with two independent steps. This assumption is ubiquitous in network science. It is at the root of node-ranking and community detection algorithms[2–5], of scalable techniques to calculate shortest paths, optimal flows and cuts[6], as well as of visualization methods[7].

The success of network models across the sciences rests on their ability to connect the structure, dynamics and function of arbitrary systems on the basis of abundant data on pairwise interactions between their components. Compared with mean-field approaches, where the interactions between all elements are summarized through a single averaged field, network models often have greater explanatory power because they account for the sparse and non-random topologies of social, biological and technological systems[1]. However, new forms of high-dimensional and time-resolved data have now shed light on the limitations of these models.

Rich data indicate who interacts with whom, but also what different types of interactions exist, when and in which order they occur, and whether interactions involve pairs or larger sets of nodes. These seemingly disparate types of data have something in common: they provide us with information on higher-order dependencies between the components of a system, which lay beyond the reach of models that exclusively capture pairwise links. This has profound consequences for network models of relational data — a cornerstone in the interdisciplinary study of complex systems. For example, higher-order dependencies have been shown to either speed up or slow down dynamical processes[8–10], change node rankings[11–13] and alter community structures[7,13,14–18].

An active community of researchers is developing higher-order network models that account for different types of higher-order dependencies in data on complex systems. Such models better capture how the components of complex systems directly and indirectly influence each other, promising improved explanatory power at the expense of increased model complexity. Further progress will require integrative approaches that combine novel network-analytic methods for rich data with scalable statistical inference and machine-learning techniques. These will help address open questions, such as finding models that optimally balance under- and overfitting in dependence of available data, or establishing the existence and scalability of a single framework that can capture different types of higher-order dependencies. Finding good answers can in turn further improve our understanding of the structure, dynamics and function of complex systems.

In this Perspective, after a brief overview of different classes of higher-order network models, we focus on the consideration of non-Markovian paths in time-stamped interaction data. We illustrate how that affects fundamental network science methods applied across disciplines, namely community detection, node ranking and the modelling of dynamical processes. Finally, we discuss challenges in developing optimal higher-order models that take advantage of rich data on higher-order dependencies while avoiding the risk of overfitting.

Modelling higher-order dependencies in complex systems

Recent work on higher-order network models can be divided into three different yet related lines of research. The first line challenges the assumption that the influence between a system's components can be decomposed into links of a single type, introducing instead multilayer higher-order models with multiple link types[1,3]. The second line questions the assumption that the influence between components in a complex system can be decomposed into pairwise links, developing models that generalize pairwise links to arbitrary node sets, which we refer to as combinatorial higher-order models[14,20–22]. The third line challenges the idea that the indirect influence between the components of a system can be understood based on transitive paths formed by independent links. Leveraging information on real paths inferred from time-series data, this research has introduced non-Markovian higher-order network models. They account for correlations in the sequence of nodes traversed by paths that cannot be captured by first-order Markov models[16,17].

[1]Mathematical Institute, University of Oxford, Oxford, UK. [2]Integrated Science Lab, Department of Physics, Umeå University, Umeå, Sweden. [3]Data Analytics Group, Department of Informatics (IFI), University of Zurich, Zurich, Switzerland. *e-mail: scholtes@ifi.uzh.ch

Notes:

- These works point to interest new research in statistical network analysis and machine learning. Considering the growing amount of time series data, we should see network models as maximum entropy models that cannot explain the complex causal paths by which the elements in a complex system indirectly influence each other. In a nutshell, the temporal dynamics (and chronological ordering) of time-stamped links in temporal networks adds another dimension of complexity to network analysis.
- We need statistical learning techniues to learn optimal higher-order models from time series data on networks. As you will see in the last exercise, the software package `pathpy`, which we develop in our group, can help you to address this challenge. It can be used to learn optimal higher-order models from time series data, and to generalise network analytic methods to such higher-order models.
- If you are interested in a broader perspective on this research area, you can find a high-level summary in the recent overview article → R Lambiotte et al. 2019

Thank you!

I count on you

- consider our follow-up course on **Machine Learning for Networks** in summer semester 2023
- contact us for **open MSc thesis topics** in network analysis and machine learning
- we have open positions at our chair (e.g. student assistants)

image credit: adaption of image from Wikimedia Commons, public domain

Notes:

- I hope that you enjoyed this year's course on statistical network analysis. I also hope that you appreciate the power of the network perspective that we have introduced in the past 13 weeks. As usual, with great power comes great responsibility. You should not blindly apply network analytic methods to relational data. Always be aware of the limitations of network models, especially when dealing with complex relational data that feature multiple types of relations, temporal characteristics, or noise.
- I count on a new generation of data and network scientists (that is: you!) that will further improve network-based data analytics and machine learning techniques.
- Thank you for being a great audience despite the special Corona-induced circumstances. It was fun to teach this course and to explore network analysis jointly with you. If you have any questions, feel free to contact me via E-mail. For those of you interested in continuing in this direction: We are happy to assign Master thesis topics in this area and we would be happy if you consider our course *Machine Learning for Networks* next summer. This new course will directly build on this course.
- We wish you success for the final exam coming up in two weeks and I hope to see you in the repetitorium, which will be held next week (more details on the time via WueCampus).

Questions

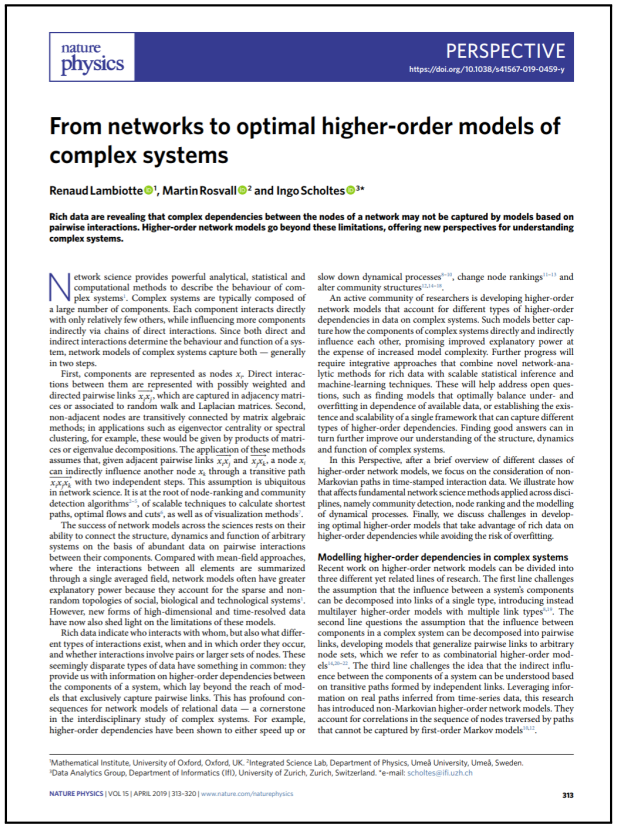
1. When is important to consider network dynamics in the study of dynamical processes on networks with changing topologies?
2. What is a causal path in a temporal network and how is it affected by the chronological ordering of time-stamped links?
3. How does the parameter δ influence the definition of a causal path.
4. Give an example for two different temporal networks that give rise to the same weighted time-aggregated network.
5. What is a second-order network? Construct the second-order network for the two temporal networks above and check whether they are different.
6. How can we define a shortest causal path in a temporal network?
7. Give an example for a temporal network where shortest causal paths are different than shortest topological paths.
8. Explain why the breaking of transitivity due to the temporal ordering of interactions can speed up diffusion, despite reducing the number of causal paths.

Notes:

References

reading list

- R Lambiotte, M Rosvall, I Scholtes: **From Networks to Optimal Higher-Order Models of Complex Systems**, Nature Physics, 2019
- F Battiston et al.: **Networks beyond pairwise interactions: structure and dynamics**, arXiv:2006.01764, 2020
- L Torres, AS Belvins, DS Basset, T Eliassi-Rad: **The why, how, and when of representations for complex systems**, arXiv:2006.02870, 2020
- P Holme: **Modern temporal network theory: a colloquium**, EPJ B, 2015
- H Lentz, T Selhorst, IM Sokolov: **Unfolding accessibility provides a macroscopic approach to temporal networks**, Phys. Rev. Lett. 2013
- R Pfitzner, I Scholtes, A Garas, CJ Tessone , F Schweitzer: **Betweenness Preference: Quantifying Correlations in the Topological Dynamics of Temporal Networks**, Phys. Rev. Lett., 2013
- I Scholtes, N Wider, R Pfitzner, A Garas, CJ Tessone, F Schweitzer: **Causality-driven slow-down and speed-up of diffusion in non-Markovian temporal networks**, Nature Comm., 2014
- H Habiba, C Tantipathananandh, and T Berger-Wolf: **Betweenness centrality measure in dynamic networks**, 2007
- I Scholtes, N Wider, A Garas: **Higher-Order Aggregate Networks in the Analysis of Temporal Networks: Path structures and centralities**, EPJ B, 2016
- I Scholtes: **When is a Network a Network? Multi-Order Graphical Model Selection in Pathways and Temporal Networks**, Proc. of SIGKDD, 2017
- V Perri, I Scholtes: **HOTVis: Higher-Order Time-Aware Visualisation of Dynamic Graphs**, Graph Drawing, 2020



nature physics

PERSPECTIVE

https://doi.org/10.1038/s41567-019-0459-y

From networks to optimal higher-order models of complex systems

Renaud Lambiotte[1], Martin Rosvall[2] and Ingo Scholtes[3]*

Rich data are revealing that complex dependencies between the nodes of a network may not be captured by models based on pairwise interactions. Higher-order network models go beyond these limitations, offering new perspectives for understanding complex systems.

Network science provides powerful analytical, statistical and computational methods to describe the behaviour of complex systems[1]. Complex systems are typically composed of a large number of components. Each component interacts directly with only relatively few others, while influencing more components indirectly via chains of direct interactions. Since both direct and indirect interactions determine the behaviour and function of a system, network models of complex systems capture both — generally in two steps.

First, components are represented as nodes x_i. Direct interactions between them are represented with possibly weighted and directed pairwise links $\overrightarrow{x_i x_j}$, which are captured in adjacency matrices or associated to random walk and Laplacian matrices. Second, non-adjacent nodes are transitively connected by matrix algebraic methods; in applications such as eigenvector centrality or spectral clustering, for example, these would be given by products of matrices or eigenvalue decompositions. The application of these methods assumes that, given adjacent pairwise links $\overrightarrow{x_i x_j}$ and $\overrightarrow{x_j x_k}$, a node x_i can indirectly influence another node x_k through a transitive path $\overrightarrow{x_i x_j x_k}$ with two independent steps. This assumption is ubiquitous in network science. It is at the root of node-ranking and community detection algorithms[2–5], of scalable techniques to calculate shortest paths, optimal flows and cuts[6], as well as of visualization methods[7].

The success of network models across the sciences rests on their ability to connect the structure, dynamics and function of arbitrary systems on the basis of abundant data on pairwise interactions between their components. Compared with mean-field approaches, where the interactions between all elements are summarized through a single averaged field, network models often have greater explanatory power because they account for the sparse and non-random topologies of social, biological and technological systems[8]. However, new forms of high-dimensional and time-resolved data have now also shed light on the limitations of these models.

Rich data indicate who interacts with whom, but also what different types of interactions exist, when and in which order they occur, and whether interactions involve pairs or larger sets of nodes. These seemingly disparate types of data have something in common: they provide us with information on higher-order dependencies between the components of a system, which lay beyond the reach of models that exclusively capture pairwise links. This has profound consequences for network models of relational data — a cornerstone in the interdisciplinary study of complex systems. For example, higher-order dependencies have been shown to either speed up or slow down dynamical processes[9–10], change node rankings[11–13] and alter community structures[12,14–18].

An active community of researchers is developing higher-order network models that account for different types of higher-order dependencies in data on complex systems. Such models better capture how the components of complex systems directly and indirectly influence each other, promising improved explanatory power at the expense of increased model complexity. Further progress will require integrative approaches that combine novel network-analytic methods for rich data with scalable statistical inference and machine-learning techniques. These will help address open questions, such as finding models that optimally balance under- and overfitting in dependence of available data, or establishing the existence and scalability of a single framework that can capture different types of higher-order dependencies. Finding good answers can in turn further improve our understanding of the structure, dynamics and function of complex systems.

In this Perspective, after a brief overview of different classes of higher-order network models, we focus on the consideration of non-Markovian paths in time-stamped interaction data. We illustrate how that affects fundamental network science methods applied across disciplines, namely community detection, node ranking and the modelling of dynamical processes. Finally, we discuss challenges in developing optimal higher-order models that take advantage of rich data on higher-order dependencies while avoiding the risk of overfitting.

Modelling higher-order dependencies in complex systems

Recent work on higher-order network models can be divided into three different yet related lines of research. The first line challenges the assumption that the influence between a system's components can be decomposed into links of a single type, introducing instead multilayer higher-order models with multiple link types[4,5]. The second line questions the assumption that the influence between components in a complex system can be decomposed into pairwise links, developing models that generalize pairwise links to arbitrary node sets, which we refer to as combinatorial higher-order models[9,20–22]. The third line challenges the idea that the indirect influence between the components of a system can be understood based on transitive paths formed by independent links. Leveraging information on real paths inferred from time-series data, this research has introduced non-Markovian higher-order network models. They account for correlations in the sequence of nodes traversed by paths that cannot be captured by first-order Markov models[10,12].

[1]Mathematical Institute, University of Oxford, Oxford, UK. [2]Integrated Science Lab, Department of Physics, Umeå University, Umeå, Sweden. [3]Data Analytics Group, Department of Informatics (IFI), University of Zurich, Zurich, Switzerland. *e-mail: scholtes@ifi.uzh.ch

Notes: