

Hands-on Tutorial on

Higher-order Data Analytics for Temporal Network Data

Welcome Note

KDD'18, London, UK

22/08/2018



Higher-order dependencies in networks

higher-order
dependencies



graph
representation

Tutors



Daniel Edler
Department of Physics
Umeå University
Sweden



Ingo Scholtes
Data Analytics Group
University of Zurich
Switzerland



Tutorial schedule

Block I	Higher-Order Network Analysis with pathpy
08:30 - 09:30	Introduction to Higher-Order Models
09:30 - 10:00	KDD Coffee Break
10:00 - 12:00	Multi-order Representation Learning
12:00 - 13:30	KDD Lunch Break
Block II	Higher-Order Clustering with InfoMap
13:30 - 15:30	Introduction to MapEquation and InfoMap
15:30 - 16:00	KDD Coffee Break
16:00 - 17:00	Higher-Order Clustering and Visualisation
Block III	Higher-Order Analysis with HONVis
online	see webpage for details

schedule and material at

<https://ingoscholtes.github.io/kdd2018-tutorial/>

Live coding



4h+ of live coding in python

```
git clone --depth 1 https://github.com/IngoScholtes/kdd2018-tutorial
```

setup instructions available at

<https://ingoscholtes.github.io/kdd2018-tutorial/setup>

python + jupyter

The data analysis and modelling framework outlined in these works builds on a generalisation of standard, first-order networks to k -dimensional De Bruijn graph models for paths in complex networks.

The class `HigherOrderNetwork` allows us to generate such higher-order network models of paths. In the documentation, we find that the constructor takes a parameter `paths`, i.e. the statistics of the observed paths that we want to model. With the parameter `k` we specify the order k of the higher-order model that we want to fit. To understand this better, let us do this for our toy example.

TODO: Read the toy example from unit 1.2 from the file `data/toy_paths.ngram`, generate a first-order model instance `hon_1` and print a summary of the resulting instance.

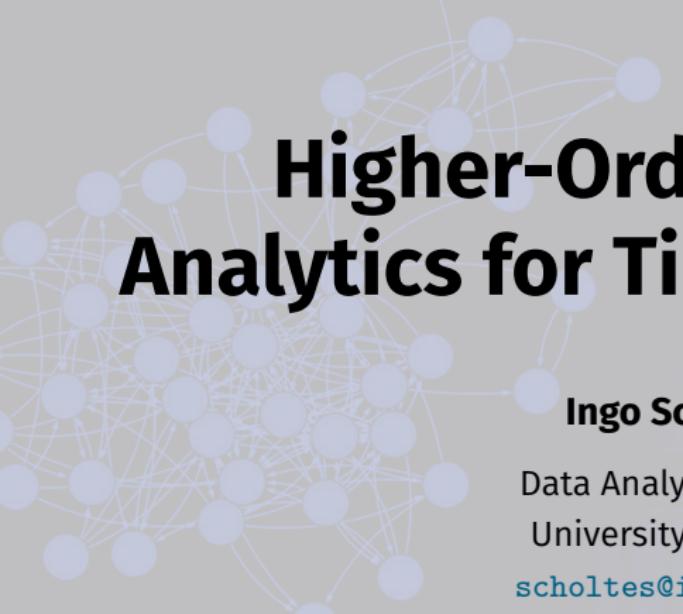
In [2]:

This generates a first-order model of our paths, with five nodes a, b, c, d and e , and four links $(a, c), (b, c), (c, d), (c, e)$. It is identical to the `Network` instance that we have previously created using `Network.from_paths`. Indeed, each `HigherOrderNetwork` instance is derived from the class `Network`, which means we can store edge and node attributes and visualise it by exactly the same methods.

TODO: Plot the `HigherOrderModel` instance `hon_1` and print the weight of all edges.

In [3]:

tutorial transcript with solutions
will be pushed to repository



Higher-Order Network Analytics for Time Series Data

Ingo Scholtes

Data Analytics Group
University of Zurich
scholtes@ifi.uzh.ch

[@ingo_s](https://twitter.com/ingo_s)

Introductory Lecture

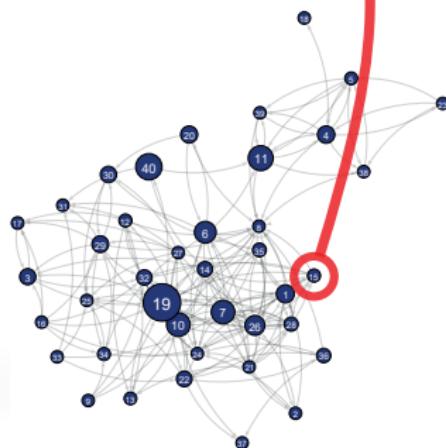
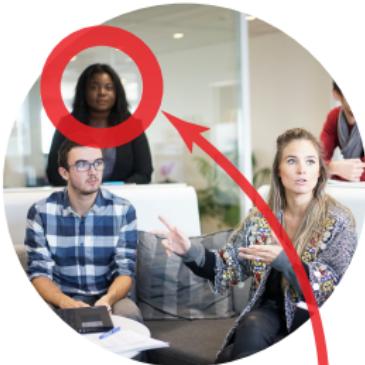
KDD'18, London, UK

22/08/2018

Mining and learning with graphs?

from → to when

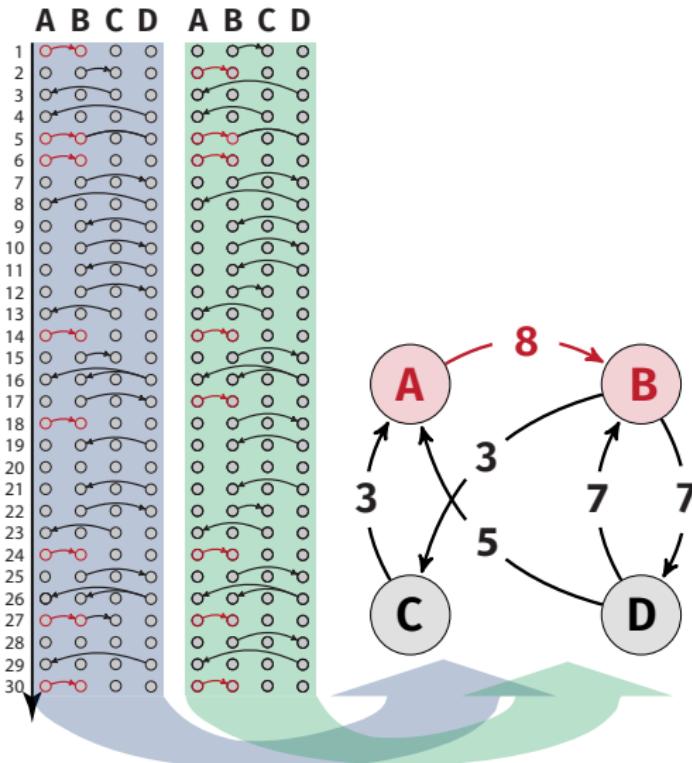
10	→	15	14:12:21
43	→	38	14:12:47
22	→	57	14:12:49
19	→	35	14:13:05
2	→	25	14:13:12
48	→	31	14:13:29
30	→	21	14:13:33
8	→	7	14:14:01
55	→	17	14:14:17
20	→	27	14:14:57
3	→	40	14:15:03
51	→	5	14:15:07
4	→	24	14:16:34
56	→	5	14:16:35
3	→	57	14:18:24
31	→	11	14:18:28
1	→	6	14:18:32
40	→	3	14:18:58
31	→	48	14:20:00
25	→	5	14:22:23
21	→	43	14:24:55
41	→	30	14:27:02
11	→	31	14:29:46
5	→	56	14:30:01
35	→	2	14:31:04
15	→	10	14:31:29



problem

- ▶ temporal information can **invalidate graph representations**
- ▶ **limits graph mining applications** in social, technical, and biological systems

Why is time important?



Ordering matters!

- ▶ graph analytic & algebraic methods:
transitive & Markovian paths, i.e.
 $P(\overrightarrow{ABC}) = P(\overrightarrow{AB}) \cdot P(\overrightarrow{BC})$
- ▶ temporal correlations lead to
non-Markovian paths
- ▶ real time series data exhibit
higher-order dependencies

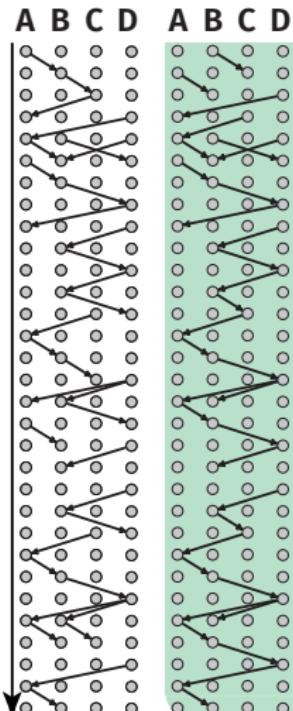
misleading results about ...

- ▶ path structures
- ▶ random walks
- ▶ cluster structures
- ▶ node centralities

→ R Pfisterer, I Scholtes et al., Phys Rev Lett 2013
→ I Scholtes et al., Nature Comm. 2014
→ M Rosvall et al., Nature Comm. 2014
→ I Scholtes et al., SIGKDD 2017



From graphs to higher-order models

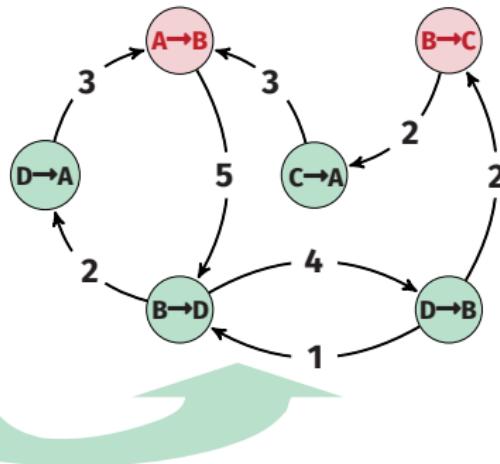


higher-order network model

k-dimensional **De Bruijn graph model**
of causal paths

→ I Scholtes et al., Nature Comm. 2014

→ memory networks, M Rosvall et al., Nature Comm. 2014



Time-aware node ranking

centrality measures

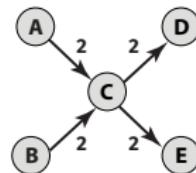
- ▶ find “influential” individuals
 - ▶ spectral centralities
 - ▶ path-based centralities
- ▶ but: focus on **topological** importance

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, EPJ B 2016

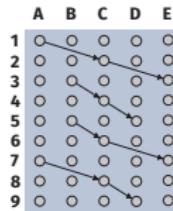
in practice

time-stamped interactions between software developers in OpenSource project Eclipse



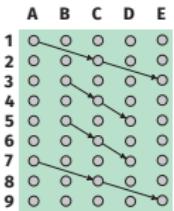
first-order node centralities

$$\text{betweenness}(C) = 4$$



time series 1

$$\text{betweenness}(C) = 4$$



time series 2

$$\text{betweenness}(C) = 2$$

Time-aware graph clustering

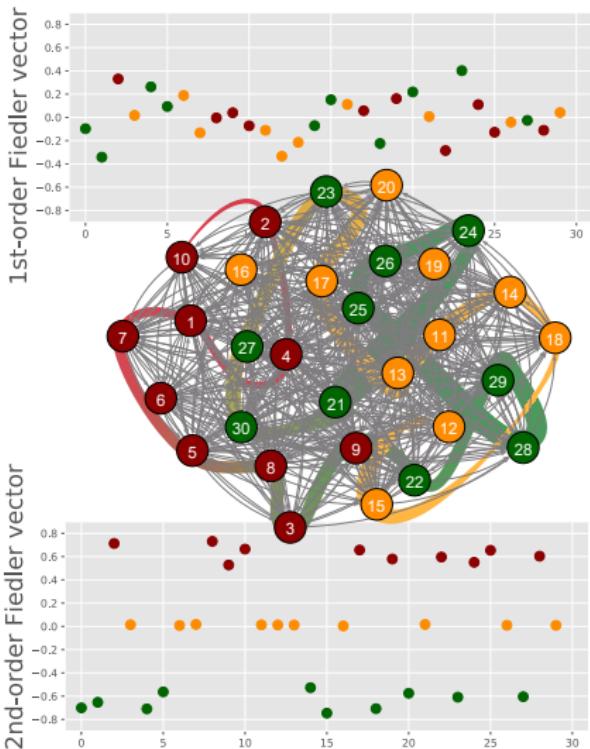
graph clustering algorithms

- ▶ 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., Nature Comm. 2014
- ▶ **higher-order flow compression**

→ D Edler, L Bohlin, M Rosvall, Algorithms, 2017



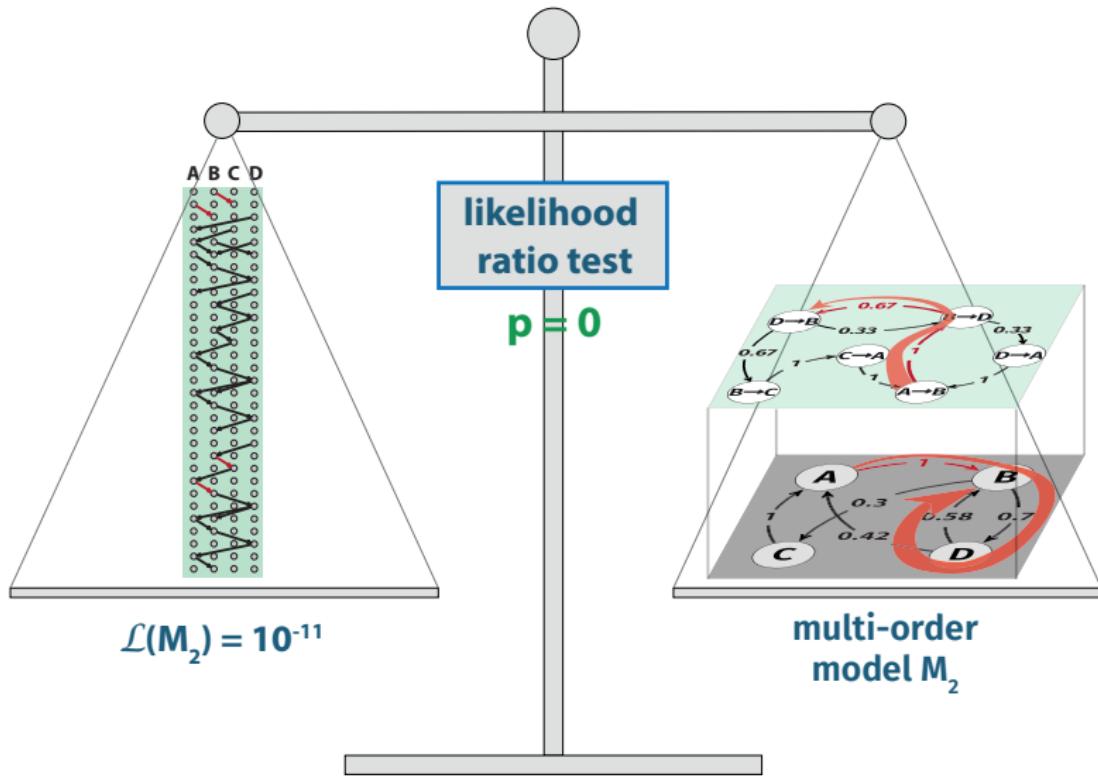
Higher-order network analytics

helps us to ...

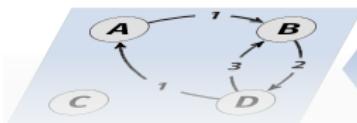
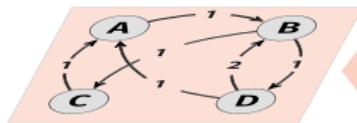
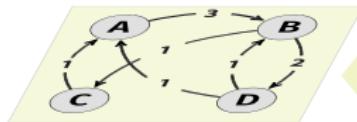
- ▶ analytically understand **flow and spreading processes** in dynamic network topologies
- ▶ **detect temporal-topological clusters** in time series data on networks
- ▶ **recognize anomalies** in temporal and sequential data
- ▶ **identify influential individuals** in time-resolved social networks

→ **live coding sessions**

Representation learning

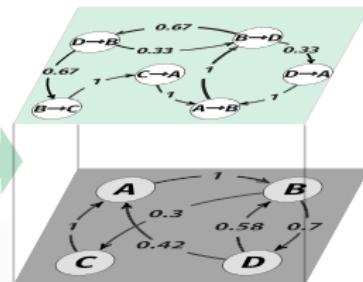


Temporal network analysis



A B C D

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



time-slice graphs

time-aggregated time slices

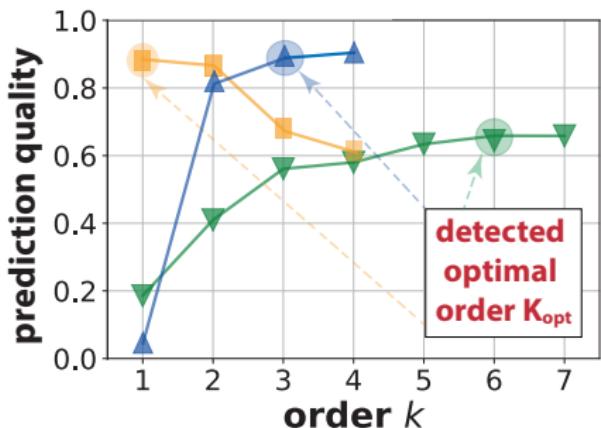
discard information on ordering

our perspective

optimal **multi-order graph**

summarisation of **causal topologies**

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, SIGKDD 2017

From networks to optimal models

Understanding Complex Systems: From Networks to Optimal Higher-Order Models

Renaud Lambiotte¹, Martin Rosvall^{*}, Ingo Scholtes²

¹ University of Oxford, United Kingdom

^{*} Integrated Science Lab, Umeå University, Sweden

² Chair of Systems Design, ETH Zürich, Switzerland

Abstract

To better understand the structure and function of complex systems, researchers often represent them as networks, in which components are nodes and interactions between them are paths. Such network models assume that actual paths are memoryless. That is, the way a path continues as it passes through a node does not depend on where it came from. Recent studies of data on actual paths in complex systems question this assumption and instead indicate that memory in paths does have considerable impact on central methods in network science. A growing research community working with so-called higher-order network models addresses this issue, seeking to take advantage of information that conventional network representations disregard. Here we summarise the progress in this area and outline remaining challenges calling for more research.

A long-standing goal of statistical physics is to understand emergent phenomena in complex systems that consist of a large number of interacting components. Such systems not only occur in condensed matter physics, but they are also widespread in other disciplines, and physicists have been able to contribute to a better understanding of biological, social, economic, and technological systems. A salient feature of complex systems is that all system components can influence each other, either directly or indirectly. Systems for which the topology of these interactions is unknown are often studied using mean-field approaches, which summarise interactions between all elements with a single averaged field. Over the past few decades, a surge of data has demonstrated that complex systems in the real world exhibit sparse and complex topologies in which few components directly interact with each other, while most components indirectly influence each other via sequences of multiple direct interactions [1]. Such systems can be conveniently represented as graphs or networks, where nodes x_i represent the components of a system and links $\overrightarrow{x_i x_j}$ capture the topology of direct pairwise interactions. The indirect influence between two components x_0 and x_n can be studied based on sequences of direct interactions $\overrightarrow{x_0 x_1}, \overrightarrow{x_1 x_2}, \dots, \overrightarrow{x_{n-1} x_n}$ that mediate the influence between x_0 and x_n via a path $\overrightarrow{x_0 x_1 \dots x_n}$.

Building on this abstraction, network science has developed methods that help us to better understand the structure and function of complex systems. The success and popularity of these network science methods across disciplines rest on their broad applicability to relational data that capture pairwise interactions. However, the analysis of such data based on linear transformations, algebraic methods, and Markovian models of dynamical processes [2, 3] also makes an important assumption, namely that the paths by which a system's components indirectly influence each other can be understood based on the *transitive closure* of direct, pairwise interactions. That is, most network science methods rest on the assumption that the existence of direct interactions $\overrightarrow{x_0 x_1}$ and $\overrightarrow{x_1 x_2}$ implies that x_0 can indirectly influence x_2 via a transitive path $\overrightarrow{x_0 x_1 x_2}$. Notably, methods based on the composition of linear transformations that capture network topologies implicitly introduce this fundamental hypothesis about indirect influence in a complex system. Examples include algebraic and spectral methods based on eigenvalues, products, and powers of adjacency matrices and Laplacians;

temporal
data on
networks

PATHPY

<http://www.pathpy.net>

optimal model
of causal
topology

arXiv:1806.05977