

3 Übung zu Complex Networks

Der **Durchmesser** eines Netzwerks ist die Länge des längsten kürzesten Pfades.

Der **durchschnittliche Abstand** $l = \langle L_{ij} \rangle$ ist die durchschnittliche Länge aller (average shortest path length) $= \langle d_{ij} \rangle$ kürzesten Pfade.

$$l = \frac{1}{N(N-1)} \sum_{\substack{i,j \\ i \neq j}} d_{ij}$$

Berechnung von l :

Beobachtung: A enthält Einträge a_{ij} , die kürzesten Pfade der Länge 1 entsprechen

$$\underline{A^2} : (a_{ij})^{(2)} = \sum_{k=1}^N a_{ik} a_{kj} \quad \text{Pfade: } j \rightarrow k \rightarrow i \quad \text{Pfade der Länge 2 von } j \rightarrow i \text{ (über } k)$$

$$\underline{A^n} : (a_{ij})^{(n)} = \# \text{Pfade der Länge } n \text{ von } j \text{ nach } i$$

$$\Rightarrow d_{ij} = \text{Potenz von } A, \text{ für die } (a_{ij})^{(n)} \text{ erstmals ungleich Null ist}$$

↑
Abstand, Länge des kürzesten Pfades

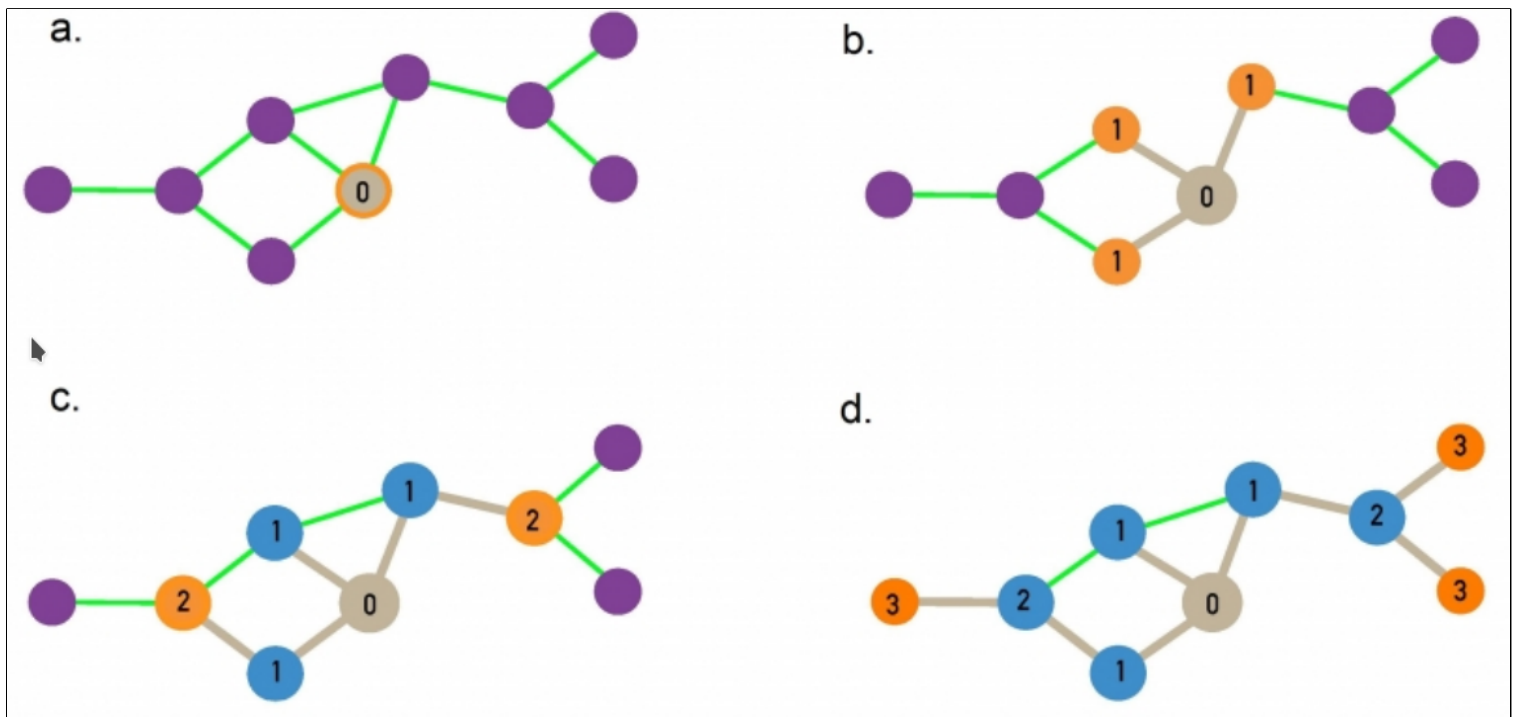
Rechenaufwand für A^2 ist $O(N^3) \Rightarrow$ kostenintensiv

geschickter Ideefür die Berechnung der kürzesten Pfade: Breadth first search **BFS** (Breitensuche)

- 1) Starte mit Knoten i (Etikett $n=0$)
- 2) Suche alle Nachbarn von i (Etikett $n=1$)
- 3) Nimm alle Knoten ohne Etikett, die benachbart zu etikettierten Knoten sind
- 4) Wiederhole Etikettierung, bis alle Nachbarn von n -etikettierten Knoten etikettiert sind. (Etikett $n+1$)
- 5) Wiederhole 3)+4) für alle $n+1$ -etikettierten Knoten, bis alle Knoten

ein Etikett haben

=> Etikett = Abstand von i



Aufwand: Jeder Knoten wird einmal etikettiert. => $O(N^2)$

Hinweis zu Aufgabe 7: Netzwerke aus Daten

Network	N	L	$\langle k \rangle$	$\langle d \rangle$	d_{\max}
Internet	192,244	609,066	6.34	6.98	26
WWW	325,729	1,497,134	4.60	11.27	93
Power Grid	4,941	6,594	2.67	18.99	46
Mobile-Phone Calls	36,595	91,826	2.51	11.72	39
Email	57,194	103,731	1.81	5.88	18
Science Collaboration	23,133	93,437	8.08	5.35	15
Actor Network	702,388	29,397,908	83.71	3.91	14
Citation Network	449,673	4,707,958	10.43	11.21	42
E. Coli Metabolism	1,039	5,802	5.58	2.98	8
Protein Interactions	2,018	2,930	2.90	5.61	14

Beispiele für vorgefertigte Routen

NetworkX

Software for complex networks

Stable (notes)

2.1 — January 2018
[download](#) | [doc](#) | [pdf](#)

Latest (notes)

2.2 development
[github](#) | [doc](#) | [pdf](#)

NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.



graph-tool | Efficient network analysis

Down

What is graph-tool?

Graph-tool is an efficient Python module for manipulation and statistical analysis of graphs (a.k.a. networks). Contrary to most other python modules with similar functionality, the core data structures and algorithms are implemented in C++, making extensive use of template metaprogramming, based heavily on the Boost Graph Library. This confers it a level of performance that is comparable (both in memory usage and computation time) to that of a pure C/C++ library.



Home Features Learn Develop Plugins Services Consortium

The Open Graph Viz Platform

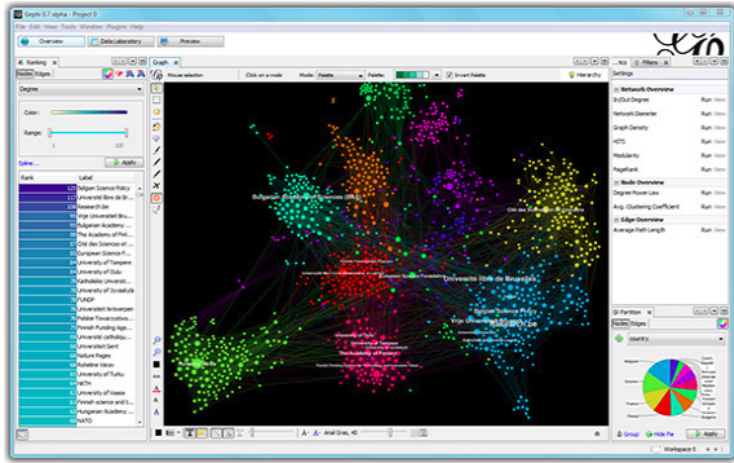
Gephi is the leading visualization and exploration software for all kinds of graphs and networks. Gephi is open-source and free.

Runs on Windows, Mac OS X and Linux.

[Learn More on Gephi Platform »](#)

[Download FREE Gephi 0.9.2](#)
[Release Notes](#) | [System Requirements](#)

- ▶ Features
- ▶ Quick start
- ▶ Screenshots
- ▶ Videos



Code-Schnipsel: siehe Internetseite zur Übung