Heidelberg University
Institute of Computer Science
Database Systems Research Group

Lecture: Complex Network Analysis

Prof. Dr. Michael Gertz

Assignment 1

Graph Theory and Networks in Python

`https://github.com/nilskre/CNA_assignments`

Team Member: Patrick Günther, TODO,
Applied Computer Science
TODO@stud.uni-heidelberg.de

Team Member: Felix Hausberger, 3661293,
Applied Computer Science
eb260@stud.uni-heidelberg.de

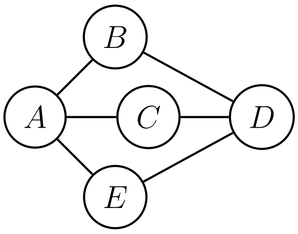
Team Member: Nils Krehl, 3664130,
Applied Computer Science
pu268@stud.uni-heidelberg.de

1 Problem 1-1 Adjacency Matrix

1. Is the corresponding graph G directed or undirected? Justify your answer.

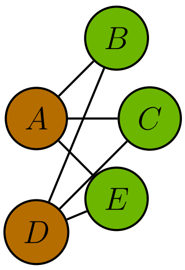
   Undirected, because the adjacency matrix is symmetric.

2. Draw the graph described by the adjacency matrix A_G. Use labels to indicate the correspondence of nodes to rows or columns of the adjacency matrix.

3. Is the graph bipartite? Explain your answer by giving 2-3 sentences.

   Yes, the graph is bipartite. It can be divided into the two disjoint sets $U = A, D$ and $V = B, C, E$. The nodes connect these two sets, but never connect two nodes in one set (independent set property).

4. Give the adjacency list and the edge list representation of the graph G.

   adjacency list:

node	linked to
A	B, C, E
B	A, D
C	A, D
D	B, C, E
E	A, D

   edge list:

pair of edges
(A, B)
(A, C)
(A, E)
(B, D)
(C, D)
(D, E)

2 Problem 1-2 Average Degree of a Growing Network

Consider the following properties of the network at time $t = T$.

1. What is the total number of nodes N?

   In each time step one node is added to the network. That is why $N = T$

2. What is the total number of links L?

   For $t = 1$ no links exist. For every node also a link is added. That is why $L = T - 1$

3. What is the average degree $\langle k \rangle$?

   The average degree is defined for an undirected graph as $\langle k \rangle = \frac{2L}{N}$. After inserting the values from above: $\langle k \rangle = \frac{2(T-1)}{T} = \frac{2T-2}{T} = 2 - \frac{2}{T}$

4. What is the average degree in the limit $T \to \infty$?

   When $T \to \infty$, the second term $\frac{2}{T}$ becomes irrelevant. Consequently the average degree becomes $\langle k \rangle = 2$ for $T \to \infty$.

3 Problem 1-3 Difficulty of an Exhaustive Search

Approximate the number of walks that need to be checked for an Eulerian trail for the following settings:

1. A complete graph with $N = 4$ nodes. Recall that a graph is complete if there is an edge between every pair of nodes in N, i.e., we have a clique.

   An undirected clique with $N = 4$ nodes has $L = \frac{N(N-1)}{2} = 6$ edges. Thus an Eulerian trail in such a graph would have a length of $l = 6$. The number of walks $W = (n_0, ..., n_l)$ that need to be checked being a potential Eulerian trail is therefore $4 \cdot 3^6 = 2.916$.

2. A regular graph with N nodes and where each node has exactly k links (that is why such a graph is called regular). The resulting formula should depend on k. By replacing k with $\langle k \rangle$ one can get an estimate for the number of walks of length K in a generic graph with N nodes and K links.

   A regular graph with N and k links per node has $L = \frac{N \cdot k}{2}$ edges. Thus an Eulerian trail in such a graph would have a length of $l = \frac{N \cdot k}{2}$. The number of walks $W = (n_0, ..., n_l)$ that need to be checked being a potential Eulerian trail is therefore $N \cdot k^{\frac{N \cdot k}{2}}$.

3. Use your results from part 2., based on $\langle k \rangle$, to compute the approximate number of walks for the bridges of Königsberg (see Figure 1).

   The average degree in the seven bridges of Königsberg problem is $\langle k \rangle = \frac{2L}{N} = 3,5$. Thus the approximate number of walks to be checked is $N \cdot \langle k \rangle^{\frac{N \cdot \langle k \rangle}{2}} = 4 \cdot 3,5^{\frac{4 \cdot 3,5}{2}} \approx 25.735,72$.

4. The historical part of Venice has 428 bridges. Assuming that $\langle k \rangle = 2$, how many walks have to be checked for being an Eulerian trail?

   The historical part of Venice has $N = \frac{2L}{\langle k \rangle} = 428$ nodes. Thus the approximate number of walks to be checked is $N \cdot \langle k \rangle^{\frac{N \cdot \langle k \rangle}{2}} = 428 \cdot 2^{\frac{428 \cdot 2}{2}} \approx 2,97 \cdot 10^{131}$.

4 Problem 1-4 Introduction to Network Processing with Python