Interactive teaching the basics of complex networks and graph mining

Documentation

Bálint Demeter, András London[*] and Tamás Németh
Institute of Informatics
University of Szeged, Hungary

July 14, 2015

1 The program

This program is user-friendly and available as a Web application at `http://www.inf.u-szeged.hu/~london/Software/graph-mining/`. It was created using JavaScript, HTML and CSS and it relies on a Web browser (front end) to render the application. JavaScript is an interpreted programming (scripting) language that is commonly used in so-called Web 2.0 platforms. One of the main advantages is that many libraries have been developed and distributed to extend the core functionality of JavaScript. The main operations like the creation and adaptation of a database using MySQL and running the algorithms (implemented in PHP) are performed on the server-side (back end).

The graph generators, the ranking algorithms, the SCC and shortest path finder algorithms were implemented in PHP. For the visualization we used the *Vis.js*, browser based visualization library. The library is designed to handle large amounts of dynamic data, and to enable the manipulation of and interaction with the data [20]. For graph clustering, we used the *D3.js* JavaScript library, which was developed for manipulating documents based on data by combining powerful visualization components and a data-driven approach to DOM manipulation [5]. In particular, we used the well known and widely used *Louvain method* [2] to explore the community structure of a graph.

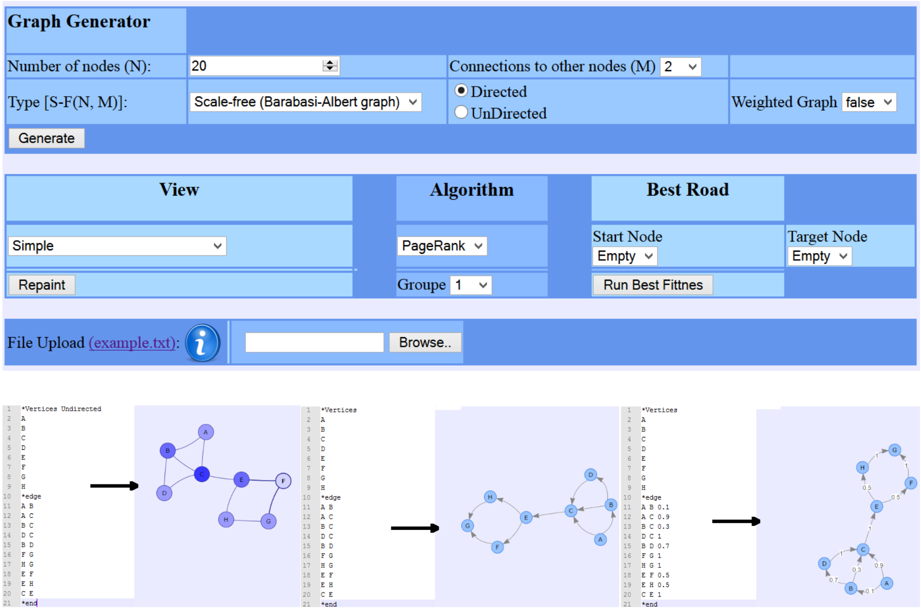
[*] Corresponding author: london@inf.u-szeged.hu

2 User's guide – step by step

Here, we give a short guide of the program in a step by step fashion.

1. Download and read the documentation if necessary
2. Generate or upload a graph (Fig. 1)
   - If generate, specify
     (a) the number of nodes
     (b) the random network model
     (c) the parameters of the model
     (d) whether the graph is directed or undirected and weighted or unweighted
   - If upload, check the format of the input graph (detailed information is available on the Web page)
3. Visualization
   - the generated or uploaded graph is displayed immediately
   - the visualized graph can be enlarged and rotated
   - the degree distribution of the network is drawn automatically
4. Change the visualization style
   (a) Simple (default)
   (b) Cluster structure (show communities)
   (c) Hierarchical layout (for directed graphs)
   (d) Strongly connected components (for directed graphs)
5. Use a ranking algorithm (for directed graphs)
   - results will be shown in a table and a diagram

   (a) PageRank – specify the damping factor and the iteration number
   (b) HITS – specify the root nodes
   (c) SALSA – specify the iteration number
6. Calculate the shortest path between two nodes
7. Download the generated graph in the specified format



Graph Generator

Number of nodes (N): 20

Connections to other nodes (M) 2

Type [S-F(N, M)]: Scale-free (Barabasi-Albert graph)

● Directed
○ UnDirected

Weighted Graph false

```
*Vertices Undirected
A
B
C
D
E
F
G
H
*edge
A B
A C
B C
D C
B D
F G
H G
E F
E H
C E
*end
```

```
*Vertices
A
B
C
D
E
F
G
H
*edge
A B
A C
B C
D C
B D
F G
H G
E F
E H
C E
*end
```

```
*Vertices
A
B
C
D
E
F
G
H
*edge
A B 0.1
A C 0.9
B C 0.3
D C 1
B D 0.7
F G 1
H G 1
E F 0.5
E H 0.5
C E 1
*end
```

Figure 1: The opening screen of the software and the input file formats for the different network types.

3 Appendix: Theoretical background

3.1 Basic notions

Formally, an *undirected (directed) graph* $G = (N, L)$ consists of two sets N and L, where $N \neq \emptyset$, while L is a set of unordered (ordered) pairs of elements of N. The elements of $N = \{1, 2, \ldots, n\}$ are called nodes and the elements of L are called links. A graph is usually represented by the *adjacency matrix* $A = [a_{ij}]_{i,j}$, which is an $n \times n$ matrix with entries $a_{ij} = 1$ if there is an edge (directed edge) between i and j and $a_{ij} = 0$ otherwise. For undirected graphs if the (i, j) edge exists, then $a_{ij} = a_{ji} = 1$, i.e. A is symmetric. If a function $w : L \to \mathbb{R}$ is given that assigns a real number to each edge, then the graph is *weighted*. The *degree* d_i of node i is the number of links that are connected to i. If the graph is directed, we can differentiate the $in-degree$ and $out-degree$ of a node i, these being $d_i^+ = \sum_n^{j=1} a_{ji}$ (the number of incoming links to i) and $d_i^- = \sum_n^{j=1} a_{ij}$ (the number of outgoing links from i), respectively.

3.2 Random-graph models of complex networks

From a graph mining perspective, random graphs can be used to investigate questions on the topological properties of typical graphs that appear in nature, social systems and technological systems. Practical applications can be found in areas where complex systems can be modeled by networks.

3.2.1 Erdős-Rényi random graphs

In mathematics, the most commonly studied random graph model was proposed by Gilbert [7]. This is the so-called $G(n, p)$ model, where n is the number of nodes, while every possible link is created independently with probability $0 < p < 1$. The degree distribution of $G(n, p)$ follows the binomial distribution; i.e. the probability that any node has degree k is

$$\Pr(d_i = k) = \binom{n}{k} p^k (1-p)^{n-k}. \tag{3.1}$$

At the same time, Erdős and Rényi studied a similar model, $G(n, M)$ [6], where a graph is chosen uniformly at random from all possible graphs with n nodes and M links. The graph $G(n, p)$ contains $p\binom{n}{2}$ links on average, a rough heuristic being that $G(n, p)$ behaves like $G(n, M)$, if $M = p\binom{n}{2}$ with increasing n and also assuming that $pn^2 \to \infty$.

The model can be easily extended to generate weighted and/or directed graphs. For historical reasons, both models are usually referred to as the Erdős-Rényi random graph model. Although it has been pointed out recently that the $G(n, p)$ model may be inappropriate for modeling real-world networks because in most real networks the average degree varies and the number of triangles is not small as in $G(n, p)$, the model is still useful for theoretical studies and making comparisons between fully random networks and real-world networks.

3.2.2 Barabási-Albert graphs

A *scale-free* network is a graph whose degree distribution follows the *power law*. That is,

$$\Pr(d_i = k) = ck^{-\gamma}, \tag{3.2}$$

where c is a constant and $\gamma > 1$.

Barabási and Albert proposed an algorithm for creating scale-free networks [1]. The model is based on a so-called *preferential attachment* mechanism. It means that nodes with more connections have a stronger capability of grabbing links added to the network. More precisely, the algorithm is the following: initially, there is an undirected and connected network with $m_0 \leq 2$ nodes,the network being connected. For each step, a new node is added to the network that connects with m new links to the nodes already present, such that the probability of attachment is proportional to their instantaneous degree. The model produces a continuously growing network, in contrast with the Erdős-Rényi random graphs, where all nodes exist initially. During the process, nodes with a higher degree get links with a higher probability, so there should be a few nodes with a lot of connections (hub nodes), but most of the nodes should have only a few connections. The extension of the model to directed graphs is straightforward.

Skewed distributions, such as power laws, occur very often and many real-world networks are scale-free. Some examples for networks that are scale-free include social networks [21], the number of sexual partners of humans [13], different kinds of computer networks including the Internet [18] and the Web-graph of the WWW [4], and airline networks [19]. In biology, important examples are the protein-protein interaction networks [8] and metabolic networks [9].

3.2.3 Watts-Strogatz graphs

A *small-world* network is a type of graph in which most of the nodes do not have many neighbors, but most nodes can be reached by only a few steps from each other node; moreover, especially in social networks, if links (i, j) and (i, k) exist, then the probability that (j, k) exists is high in general. These two properties can be formulated mathematically with the following notions. First, in small-world networks, the *average path lengths* are short (meaning that any node can be reached in just a few steps through a series of links and points from any other node), and the *clustering* is high, which can be measured using

$$C = \frac{3 \times \#\{\text{triangles}\}}{\#\{\text{connected triples of nodes}\}} \tag{3.3}$$

Watts and Strogatz introduced an algorithm ([22]) that produces a small-world graph starting from a regular graph of n nodes (where the degree of each node is the same) and rewired the edges with a certain probability.

3.3 Basic graph algorithms

First of all, two simple, but important algorithms were included in the program. These are Dijkstra's *shortest path*s finder algorithm ([16]) and Tarjan's algorithm that computes the *strongly connected components* (SCC) in a graph [17].

In the shortest path problem we have to find a path between two nodes in a graph such that the sum of the weights of the links on the path is a minimum (Fig. 2, Left). A directed graph is strongly connected if every node is reachable from every other node in a directed path. The strongly connected components of an arbitrary directed graph is a partition into subgraphs that are strongly connected (Fig. 2, Right).

3.4 Ranking actors in networks

The problem of assigning scores to a set of individuals based on their pairwise comparisons appears in many areas and activities. For example, in sports players or teams are ranked according to the outcomes of games that were played; the impact of scientific publications can be measured using the relations among their citations. Web search engines rank websites based on their hyperlink structure. The centrality of individuals in social systems can also be evaluated according to their social relations. The ranking of individuals based on the underlying

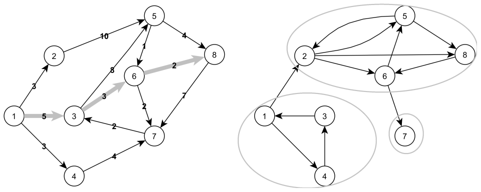


Figure 2: Left: shortest path between two nodes in a network. Right: Strongly connected components of a network.

graph that models their bilateral relations has become the central ingredient of Google's search engine and later it appeared in many areas from social network analysis to optimization in technical networks (e.g. road and electric networks) [11].

Here, we will give a brief overview on the ranking algorithms that are available in our teaching application. We should add here that all the methods can be readily extended to weighted networks and we have implemented all the algorithms in this way.

3.4.1 In–degree ranking

The simplest heuristic for ranking actors in a network is to rank them based on the number of links that point to them. Returning to the above-mentioned examples, this means that the player or team with the highest number of wins, or the scientific article with the highest number of citations will be ranked highest. However, there are at least two notable problems associated with this approach. Firstly, it can not distinguish between nodes with the same in-degree; and secondly in the evaluation of the rank of a certain node, it does not take into account how important (or the rank of) these nodes are, where the in-links come from.

3.4.2 PageRank

The key idea of Brin and Page [3] to extend the concept of in-degrees arose from the observation that not all links have the same importance. For example, a win against a highly ranked, strong opponent is more important than a win against a much weaker opponent; a hyperlink from a particular webpage does not have the same impact as a direct link from Google or Yahoo. Generally speaking, in the evaluation of the ranking of the nodes, in-links coming from highly linked nodes are more important than from nodes with just a few links (Fig 3, Left). Based on this notion, the ranking scores (PR) of the nodes are calculated iteratively as

$$PR(i) = (1-\lambda)\frac{1}{n} + \lambda \sum_{j:j\to i} \frac{PR(j)}{d_j^-}. \tag{3.4}$$

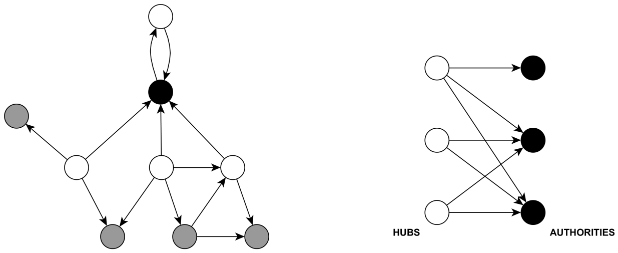


Figure 3: Left: PageRank score of the black node is calculated recursively by using the scores of the white nodes. Right: Hubs and authorities in a network.

Here, $\lambda \in [0, 1]$ is a free parameter with a value usually lying between 0.6 and 0.8. The second term simply expresses the fact that the rank of each node i is calculated according to the rank of the nodes that have a link to i, while the first term guarantees that the PR scores will converge.

3.4.3 HITS

Independent of Brin and Page, Kleinberg [10] proposed a different approach to measure the importance of a Web page. While PageRank computes the pagerank scores on the entire graph, the Kleinberg's *HITS algorithm* (Hyperlink Induced Topic Search)tries to distinguish between *hubs* and *authorities* within a subgraph of relevant pages, where hub scores and authority scores of the nodes are recursively calculated from each other. A good hub is a node that links to many authorities, while a good authority is a node that has in-coming links from good hubs (Fig. 3, Right). Mathematically, the hub and authority scores can be calculated recursively as

$$H(i) = \sum_{j:i\to j} A(j) \text{ and } A(i) = \sum_{j:j\to i} H(j) \tag{3.5}$$

respectively, and the scores will converge starting from any initial scores of the nodes.

3.4.4 SALSA

Lempel and Moran proposed an alternative algorithm [12] which combines the ideas of both PageRank and HITS and called it *SALSA* (The Stochastic Approach for Link-Structure Analysis). Like HITS, it works on a focused subgraph and calculates both hub and authority scores of the nodes. In addition, SALSA computes the different scores in a PageRank recursive manner, like so:

$$H(i) = \sum_{i\to j} \frac{1}{d_j^+} A(j), \tag{3.6}$$

$$A(i) = \sum_{j:j\to i} \frac{1}{d_j^-} H(j). \tag{3.7}$$

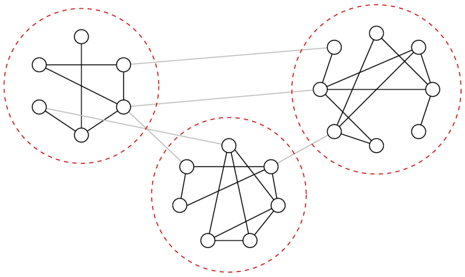


Figure 4: Network with community structure.

3.5 Communities in networks

Besides the power-low degree distribution and small-world properties (like short path lengths and high clustering), another important and common feature of complex networks is the community structure [15]. In practice, community detection in a graph is a partition of the nodes into sets, such that nodes in the same community are more densely connected to each other than to the rest of the graph (Fig. 4). In general, the communities in networks reflect the similarity and common features of the nodes that they contain.

Newman and Girvan introduced the *modularity optimization method* [14] based on the idea that a random graph is not expected to have a cluster structure like the original one. Hence, the existence of clusters is revealed by comparisons of the edge densities in certain subgraphs with the densities that would be expected if the links of the graph were randomly wired, which depend on the *null model* chosen. One of the most popular null models keeps the degree sequence of the original graph, but has the edges rewired at random, assuming that the expected degree of each node is the same as in the original graph. The modularity function that has to be optimized is defined as

$$Q = \frac{1}{2|E|}\sum_{i,j}(a_{ij} - p_{ij})\delta(C_i, C_j). \tag{3.8}$$

Here, $\delta(C_i, C_j) = 1$ if and only if $C_i = C_j$ and 0 otherwise. For an unweighted graph, $p_{ij} = \frac{d_i d_j}{2|E|}$ is the probability whose nodes i and j are connected in a random graph with the same degree sequence (null model) as the original one. An extension of the model to weighted and/or directed graphs can be easily performed.

References

[1] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[3] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

[4] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer networks*, 33(1):309–320, 2000.

[5] d3Js. Javascript library. `http://d3js.org/`, 2014.

[6] Paul Erdős and A Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.

[7] Edgar N Gilbert. Random graphs. *The Annals of Mathematical Statistics*, pages 1141–1144, 1959.

[8] Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.

[9] Hawoong Jeong, Bálint Tombor, Réka Albert, Zoltan N Oltvai, and A-L Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[10] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[11] Amy N Langville and Carl D Meyer. *Google's PageRank and beyond: The science of search engine rankings*. Princeton University Press, 2011.

[12] Ronny Lempel and Shlomo Moran. The stochastic approach for link-structure analysis (salsa) and the TKC effect. *Computer Networks*, 33(1):387–401, 2000.

[13] Fredrik Liljeros, Christofer R Edling, Luis A Nunes Amaral, H Eugene Stanley, and Yvonne Åberg. The web of human sexual contacts. *Nature*, 411(6840):907–908, 2001.

[14] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[15] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[16] S Skiena. Dijkstra's algorithm. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley*, pages 225–227, 1990.

[17] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[18] Alexei Vazquez, Romualdo Pastor-Satorras, and Alessandro Vespignani. Internet topology at the router and autonomous system level. *arXiv preprint cond-mat/0206084*, 2002.

[19] Trivik Verma, Nuno AM Araújo, and Hans J Herrmann. Revealing the structure of the world airline network. *arXiv preprint arXiv:1404.1368*, 2014.

[20] VisJs. browser based visualization library. `http://visjs.org/`, 2010.

[21] Stanley Wasserman and Joseph Galaskiewicz. *Advances in social network analysis: Research in the social and behavioral sciences*, volume 171. Sage Publications, 1994.

[22] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.