(Nonstatistical) Analysis of Large Networks

Vladimir Batagelj
University of Ljubljana

COSIN final meeting
Salou, Tarragona, Spain, March 14–18, 2005
Outline

1. Networks ................................................................. 1
2. Large Networks .......................................................... 2
3. Decompositions .......................................................... 3
4. Cuts ........................................................................... 4
5. Simple analysis using cuts ............................................. 5
6. Citation weights ......................................................... 6
7. Cores and generalized cores .......................................... 7
8. Islands ....................................................................... 8
14. Triangular connectivity and triangular networks ............ 14
18. Pattern searching ....................................................... 18
Networks

A network \( \mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W}) \) consists of:

- a graph \( \mathcal{G} = (\mathcal{V}, \mathcal{L}) \), where \( \mathcal{V} \) is the set of vertices, \( \mathcal{A} \) is the set of arcs, \( \mathcal{E} \) is the set of edges, and \( \mathcal{L} = \mathcal{E} \cup \mathcal{A} \) is the set of links. \( n = \text{card}(\mathcal{V}) \), \( m = \text{card}(\mathcal{L}) \)

- \( \mathcal{P} \) vertex value functions / properties: \( p : \mathcal{V} \rightarrow \mathcal{A} \)

- \( \mathcal{W} \) line value functions / weights: \( w : \mathcal{L} \rightarrow \mathcal{B} \)

In November 1996 we started the development of Pajek – a program, for analysis and visualization of large networks. The latest version of Pajek is freely available, for noncommercial use, at its home page:

http://vlado.fmf.uni-lj.si/pub/networks/pajek/
Large Networks

Large network – several thousands or millions of vertices.

Usually sparse $m \ll n^2$; typical: $m = O(n)$ or $m = O(n \log n)$.

Examples:

| network                  | size | $n = |V|$ | $m = |L|$    | source                           |
|-------------------------|------|--------|----------|----------------------------------|
| ODLIS dictionary        | 61K  | 2909   | 18419    | ODLIS online                     |
| Citations SOM           | 168K | 4470   | 12731    | Garfield’s collection            |
| Molecula 1ATN           | 74K  | 5020   | 5128     | Brookhaven PDB                   |
| Comput. geometry        | 140K | 7343   | 11898    | BiBTeX bibliographies            |
| English words 2-8       | 520K | 52652  | 89038    | Knuth’s English words            |
| Internet traceroutes    | 1.7M | 124651 | 207214   | Internet Mapping Project         |
| Franklin genealogy      | 12M  | 203909 | 195650   | Roperld.com gedcoms              |
| World-Wide-Web          | 3.6M | 325729 | 1497135  | Notre Dame Networks              |
| Actors                  | 3.9M | 392400 | 1342595  | Notre Dame Networks              |
| US patents              | 82M  | 3774768| 16522438 | Nber                             |
| SI internet             | 38M  | 5547916| 62259968 | Najdi Si                         |

Two main approaches: statistics and decompositions.
Decompositions

The main goals in the design of Pajek are:

- to support abstraction by (recursive) decomposition of a large network into several smaller networks that can be treated further using more sophisticated methods;
- to provide the user with some powerful visualization tools;
- to implement a selection of efficient subquadratic algorithms for analysis of large networks.

With Pajek we can: find clusters (components, neighbourhoods of ‘important’ vertices, cores, etc.) in a network, extract vertices that belong to the same clusters and show them separately, possibly with the parts of the context (detailed local view), shrink vertices in clusters and show relations among clusters (global view).
Cuts

The standard approach to find interesting groups inside a network was based on properties/weights – they can be measured or computed from network structure (for example Kleinberg’s hubs and authorities).

The vertex-cut of a network $N = (V, L, p)$, $p : V \rightarrow \mathbb{R}$, at selected level $t$ is a subnetwork $N(t) = (V', L(V'), p)$, determined by the set

$$V' = \{v \in V : p(v) \geq t\}$$

and $L(V')$ is the set of lines from $L$ that have both endpoints in $V'$.

The line-cut of a network $N = (V, L, w)$, $w : V \rightarrow \mathbb{R}$, at selected level $t$ is a subnetwork $N(t) = (V(L'), L', w)$, determined by the set

$$L' = \{e \in L : w(e) \geq t\}$$

and $V(L')$ is the set of all endpoints of the lines from $L'$. 
Simple analysis using cuts

We look at the components of $N(t)$.

Their number and sizes depend on $t$. Usually there are many small components. Often we consider only components of size at least $k$ and not exceeding $K$. The components of size smaller than $k$ are discarded as 'noninteresting'; and the components of size larger than $K$ are cut again at some higher level.

The values of thresholds $t$, $k$ and $K$ are determined by inspecting the distribution of vertex/arc-values and the distribution of component sizes and considering additional knowledge on the nature of network or goals of analysis.

We developed some new and efficiently computable properties/weights.
The citation network analysis started in 1964 with the paper of Garfield et al. In 1989 Hummon and Doreian proposed three indices – weights of arcs that are proportional to the number of different source-sink paths passing through the arc. We developed algorithms to efficiently compute these indices.

Main subnetwork (arc cut at level 0.007) of the SOM (self-organizing maps) citation network (4470 vertices, 12731 arcs).

See paper.
The notion of core was introduced by Seidman in 1983. Vertices belonging to a $k$-core have to be linked to at least $k$ other vertices of the core. A very efficient algorithm exists for determining cores.

The notion of core can be extended to other vertex functions and for several of them the corresponding cores can be efficiently determined.

Figure presents the $p_S$-core at level 46 of the collaboration network (7343 vertices, 11898 edges, edge weight counts the number of common works) in the field of computational geometry.

See paper.
Islands

If we represent a given or computed value of vertices / lines as a height of vertices / lines and we immerse the network into a water up to selected level we get islands. Varying the level we get different islands. Islands are very general and efficient approach to determine the ’important’ subnetworks in a given network.

We developed very efficient algorithms to determine the islands hierarchy and to list all the islands of selected sizes.

See details.
Islands - Reuters terror news

Using CRA S. Corman and K. Dooley produced the Reuters terror news network that is based on all stories released during 66 consecutive days by the news agency Reuters concerning the September 11 attack on the US. The vertices of a network are words (terms); there is an edge between two words iff they appear in the same text unit. The weight of an edge is its frequency. It has $n = 13332$ vertices and $m = 243447$ edges.
Islands – US patents

As an example, let us look at Nber network of US Patents. It has 3774768 vertices and 16522438 arcs (1 loop). We computed SPC weights in it and determined all (2,90)-islands. The reduced network has 470137 vertices, 307472 arcs and for different $k$: $C_2 = 187610$, $C_5 = 8859$, $C_{30} = 101$, $C_{50} = 30$ islands. Rolex

\[
\begin{bmatrix}
0 & 139793 & 29670 & 9288 & 3966 & 1827 & 997 & 578 & 362 & 250 \\
190 & 125 & 104 & 71 & 47 & 37 & 36 & 33 & 21 & 23 \\
17 & 16 & 8 & 7 & 13 & 10 & 10 & 5 & 5 & 5 \\
12 & 3 & 7 & 3 & 3 & 3 & 2 & 6 & 6 & 2 \\
1 & 3 & 4 & 1 & 5 & 2 & 1 & 1 & 1 & 1 \\
2 & 3 & 3 & 2 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 7
\end{bmatrix}
\]
Island size distribution
Main path and main island of Patents
Triangular connectivity and triangular networks

We can assign to a given graph a triangular network in which every line of the original graph gets as its weight the number of triangles that contain it. The triangular weights provide us, combined with islands, with a very efficient way to identify dense parts of a graph.

These notions can be generalized to short cycle connectivity (see paper).
Edge-cut at level 16 of triangular network of Erdős collaboration graph

without Erdős,

\( n = 6926, \quad m = 11343 \)
Edge-cut at level 11 of transitive network of ODLIS dictionary graph
Islands – The Edinburgh Associative Thesaurus

\[ n = 23219, \quad m = 325624, \text{ transitivity weight} \]
Pattern searching

If a selected pattern determined by a given graph does not occur frequently in a sparse network the straightforward backtracking algorithm applied for pattern searching finds all appearances of the pattern very fast even in the case of very large networks. Pattern searching was successfully applied to searching for patterns of atoms in molecular (carbon rings) and searching for relinking marriages in genealogies.

Three connected relinking marriages in the genealogy (represented as a p-graph) of Ragusan noble families. A solid arc indicates the _ is a son of _ relation, and a dotted arc indicates the _ is a daughter of _ relation. In all three patterns a brother and a sister from one family found their partners in the same other family.