Network Analysis in Complex Environment

Analysis of weighted networks

Vladimir Batagelj

IMFM Ljubljana, IAM UP Koper

Spinaker Summer School

Kraków, Poland, 3-14. July 2023

European Funds Knowledge Education Development

Republic of Poland

NAWA Polish National Agency for Academic Exchange

European Union European Social Fund

Outline

PROGRAM SPINAKER

Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si

Last version of slides (July 5, 2023, 02 : 16): Wnets.pdf

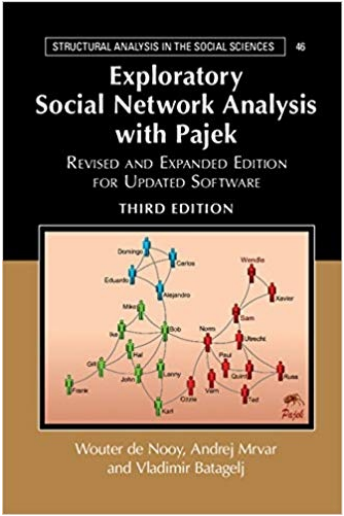
Info

Timetable

Fri	7^{th}	14:00 – 17:15
Mon	10^{th}	15:45 – 17:15
Tue	11^{th}	11:30 – 13:00
Wed	12^{th}	15:45 – 17:15
Thu	13^{th}	14:00 – 15:30

https://github.com/bavla/wNets

ESNA **Pajek**



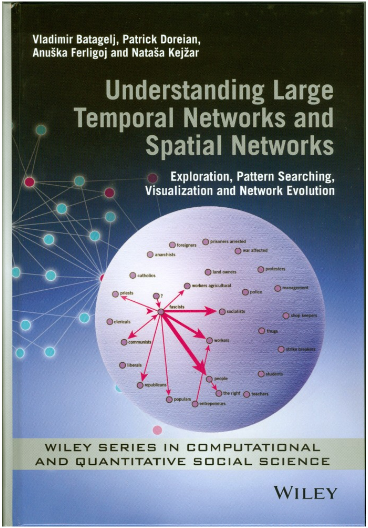
An introduction to social network analysis with **Pajek** is available in the book ESNA 3 (de Nooy, Mrvar, Batagelj, CUP 2005, 2011, 2018).

ESNA in Japanese was published by Tokyo Denki University Press in 2010; and in Chinese by Beijing World Publishing in November 2012.

Pajek – program for analysis and visualization of large networks is freely available, for noncommercial use, at its web site.

http://mrvar.fdv.uni-lj.si/pajek/

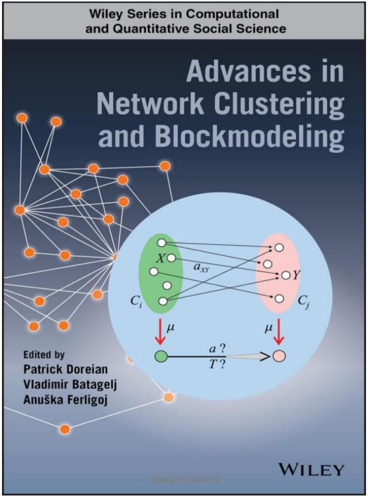
Understanding large networks



This course is closely related to chapters 2 and 3 in the book:

Vladimir Batagelj, Patrick Doreian, Anuška Ferligoj and Nataša Kejžar: Understanding Large Temporal Networks and Spatial Networks: Exploration, Pattern Searching, Visualization and Network Evolution. Wiley Series in Computational and Quantitative Social Science. Wiley, October 2014.

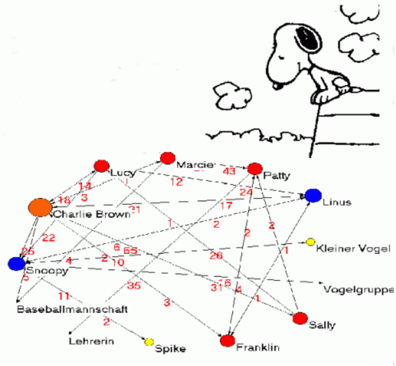
Advances in Network Clustering and Blockmodeling



An overview of the network clustering and blockmodeling is available in the book:

Patrick Doreian, Vladimir Batagelj, Anuška Ferligoj (Eds.): Advances in Network Clustering and Blockmodeling. Wiley Series in Computational and Quantitative Social Science. Wiley, 2020.

Networks



Alexandra Schuler/ Marion Laging-Glaser:
Analyse von Snoopy Comics

A *network* is based on two sets – set of *nodes* (vertices), that represent the selected *units*, and set of *links* (lines), that represent *ties* between units. They determine a *graph*. A link can be *directed* – an *arc*, or *undirected* – an *edge*.

Additional data about nodes or links can be known – their *properties* (attributes). For example: name/label, type, value, . . .

Network = Graph + Data

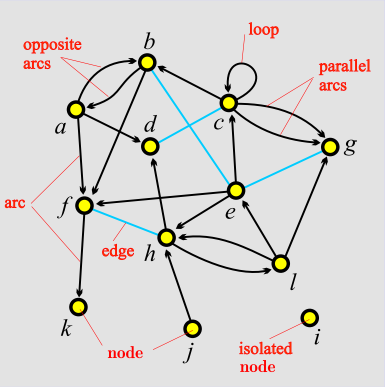
The data can be measured or computed.

Networks / Formally

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

- a *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{A}$ 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.
  $n = |\mathcal{V}|$, $m = |\mathcal{L}|$
- $\mathcal{P}$ *node value functions* / properties: $p : \mathcal{V} \to A$
- $\mathcal{W}$ *link value functions* / weights: $w : \mathcal{L} \to B$

Graph

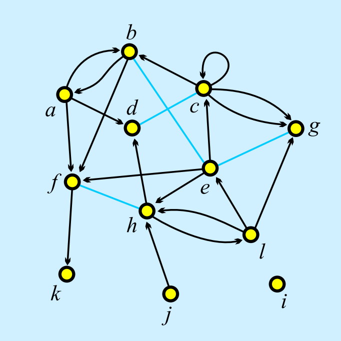


unit, actor – node, vertex
tie, line – link, edge, arc

arc $=$ directed link, (a, d)
a is the *initial* node,
d is the *terminal* node.

edge $=$ undirected link, $(c : d)$
c and d are *end* nodes.

Graph / Sets – NET

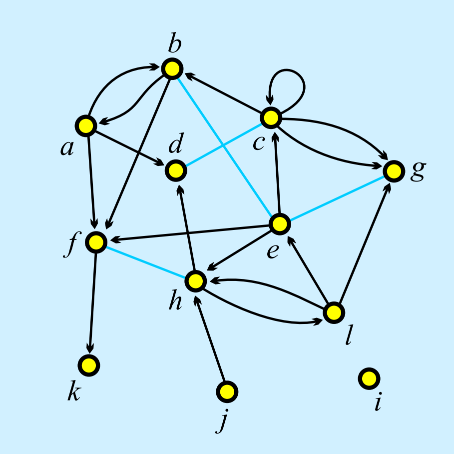


$$\begin{aligned}
\mathcal{V} &= \{a, b, c, d, e, f, g, h, i, j, k, l\} \\
\mathcal{A} &= \{(a, b), (a, d), (a, f), (b, a), \\
&\quad (b, f), (c, b), (c, c), (c, g)_1, \\
&\quad (c, g)_2, (e, c), (e, f), (e, h), \\
&\quad (f, k), (h, d), (h, l), (j, h), \\
&\quad (l, e), (l, g), (l, h)\} \\
\mathcal{E} &= \{(b: e), (c: d), (e: g), (f: h)\} \\
\mathbf{G} &= (\mathcal{V}, \mathcal{A}, \mathcal{E}) \\
\mathcal{L} &= \mathcal{A} \cup \mathcal{E}
\end{aligned}$$

$\mathcal{A} = \emptyset$ – *undirected* graph; $\mathcal{E} = \emptyset$ – *directed* graph.

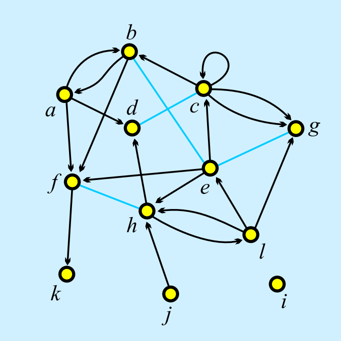
Pajek: local: `GraphSet`; `TinaSet`;
WWW: `GraphSet` / `net`; `TinaSet` / `net`, picture `picture`.

Graph / Sets – NET




```
*Vertices  12
  1 "a"     0.1020   0.3226
  2 "b"     0.2860   0.0876
  3 "c"     0.5322   0.2304
  4 "d"     0.3259   0.3917
  5 "e"     0.5543   0.4770
  6 "f"     0.1552   0.6406
  7 "g"     0.8293   0.3249
  8 "h"     0.4479   0.6866
  9 "i"     0.8204   0.8203
 10 "j"     0.4789   0.9055
 11 "k"     0.1175   0.9032
 12 "l"     0.7095   0.6475
*Arcs
  1  2
  2  1
  1  4
  1  6
  2  6
  3  2
  3  3
  3  7
  3  7
  5  3
  5  6
  5  8
  6 11
  8  4
 10  8
 12  5
 12  7
  8 12
 12  8
*Edges
  2  5
  3  4
  5  7
  6  8
```

Graph / Matrix – MAT

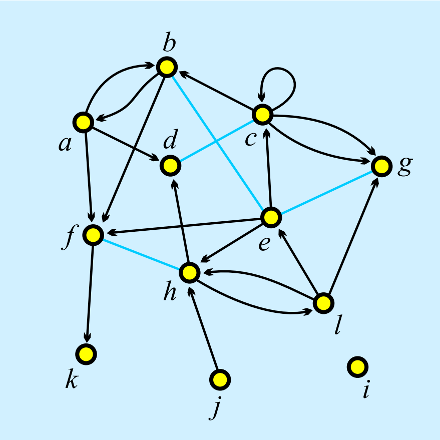


	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	0	1	0	1	0	0	0	0	0	0
b	1	0	0	0	1	1	0	0	0	0	0	0
c	0	1	1	1	0	0	2	0	0	0	0	0
d	0	0	1	0	0	0	0	0	0	0	0	0
e	0	1	1	0	0	1	1	1	0	0	0	0
f	0	0	0	0	0	0	0	1	0	0	1	0
g	0	0	0	0	1	0	0	0	0	0	0	0
h	0	0	0	1	0	1	0	0	0	0	0	1
i	0	0	0	0	0	0	0	0	0	0	0	0
j	0	0	0	0	0	0	0	1	0	0	0	0
k	0	0	0	0	0	0	0	0	0	0	0	0
l	0	0	0	0	1	0	1	1	0	0	0	0

Pajek: local: `GraphMat`; `TinaMat`, picture `picture`;
WWW: `GraphMat` / `net`; `TinaMat` / `net`, `paj`.

Graph *G* is *simple* if in the corresponding matrix all entries are 0 or 1.

Graph / Matrix – MAT




```
*Vertices   12
   1 "a"     0.1020   0.3226
   2 "b"     0.2860   0.0876
   3 "c"     0.5322   0.2304
   4 "d"     0.3259   0.3917
   5 "e"     0.5543   0.4770
   6 "f"     0.1552   0.6406
   7 "g"     0.8293   0.3249
   8 "h"     0.4479   0.6866
   9 "i"     0.8204   0.8203
  10 "j"     0.4789   0.9055
  11 "k"     0.1175   0.9032
  12 "l"     0.7095   0.6475
*Matrix
 0  1  0  1  0  1  0  0  0  0  0  0
 1  0  0  0  1  1  0  0  0  0  0  0
 0  1  1  1  0  0  2  0  0  0  0  0
 0  0  1  0  0  0  0  0  0  0  0  0
 0  1  1  0  0  1  1  1  0  0  0  0
 0  0  0  0  0  0  0  1  0  0  1  0
 0  0  0  0  1  0  0  0  0  0  0  0
 0  0  0  1  0  1  0  0  0  0  0  1
 0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  1  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  1  0  1  1  0  0  0  0
```

Node Properties / CLU, VEC, PER

All three types of files have the same structure:

`*vertices` n	n is the number of nodes
v_1	node 1 has value v_1
...	
v_n	

CLUstering – partition of nodes – *nominal* or *ordinal* data about nodes
$v_i \in \mathbb{N}$: node i belongs to the cluster/group v_i;

VECtor – *numeric* data about nodes
$v_i \in \mathbb{R}$: the property has value v_i on node i;

PERmutation – *ordering* of nodes
$v_i \in \mathbb{N}$: node i is at the v_i-th position.

When collecting the network data consider to provide as much properties as possible.

Example: Wolfe Monkey Data

inter.net

```
*Vertices  20
  1 "m01"
  2 "m02"
  3 "m03"
  4 "m04"
  5 "m05"
  6 "f06"
  7 "f07"
  8 "f08"
  9 "f09"
 10 "f10"
 11 "f11"
 12 "f12"
 13 "f13"
 14 "f14"
 15 "f15"
 16 "f16"
 17 "f17"
 18 "f18"
 19 "f19"
 20 "f20"
*Edges
  1   2   2
  1   3  10
  1   4   4
```

inter.net

```
  1   6   5
  1   7   9
  1   8   7
  1   9   4
  1  10   3
  1  11   3
  1  12   7
  1  13   3
  1  14   2
  1  15   5
  1  16   1
  1  17   4
  1  18   1
  2   3   5
  2   4   1
  2   5   3
  2   6   1
  2   7   4
  2   8   2
  2   9   6
  2  10   2
  2  11   5
  2  12   4
  2  13   3
  2  14   2
       ...
```

sex.clu

```
*vertices 20
 1
 1
 1
 1
 1
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
 2
```

age.vec

```
*vertices 20
 15
 10
 10
  8
  7
 15
  5
 11
  8
  9
 16
 10
 14
  5
  7
 11
  7
  5
 15
  4
```

rank.per

```
*vertices 20
  1
  2
  3
  4
  5
 10
 11
  6
 12
  9
  7
  8
 18
 19
 20
 13
 14
 15
 16
 17
```

Important note: 0 is not allowed as node number.

Pajek's Project File / PAJ

All types of data can be combined into a single file – Pajek's *project* file *file*.`paj`.

The easiest way to do this is:

- read all data files in Pajek,
- compute some additional data,
- delete (dispose) some data,
- save all as a project file with `File/Pajek Project File/Save`.

Next time you can restore everything with a single `File/Pajek Project File/Read`.

Wolfe network as a (Pajek's project file).

Sources of networks

Indexes and repositories

- ICON - The Colorado Index of Complex Networks
- Netzschleuder
- Network Data Repository
- Kaggle
- UCI Network Data Repository
- KONECT - Koblenz Network Collection
- Bayesys
- Stanford Large Network Dataset Collection
- Pajek datasets
-

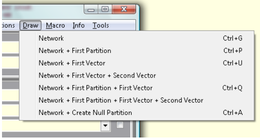
Representations of properties

Properties of nodes $\mathcal{P}$ and links $\mathcal{W}$ can be measured in different scales: numerical, ordinal and nominal. They can be *input* as data or *computed* from the network.

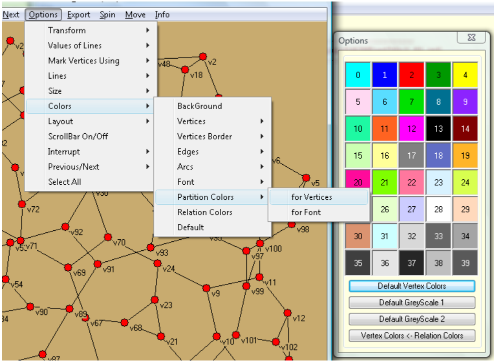
In **Pajek** numerical properties of nodes are represented by *vector*s, nominal properties by *partitions* or as *label*s of nodes. Numerical property can be displayed as *size* (width and height) of node (figure), as its *coordinate*; and a nominal property as *color* or *shape* of the figure, or as a node's *label* (content, size and color).

We can assign in **Pajek** numerical values to links. They can be displayed as *value*, *thickness* or *grey level*. Nominal vales can be assigned as *label*, *color* or *line pattern* (see Pajek manual, section ***4.3***).

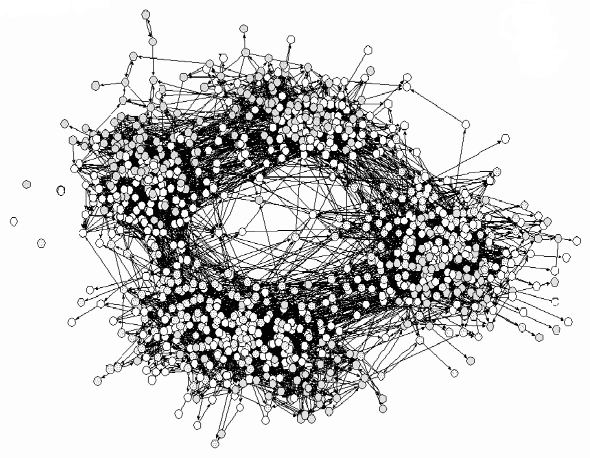
Some related operations




```
Operations/Network+Vector/Transform/Put C
Network/Create Vector/Get Coordinate
[Draw] Options
[Draw] Layout/Energy/Kamada-Kawai/Free
[Draw] Export/2D/EPS-PS
```



"Countryside" school district



James Moody (2001) AJS Vol 107, 3,679–716, friendship relation

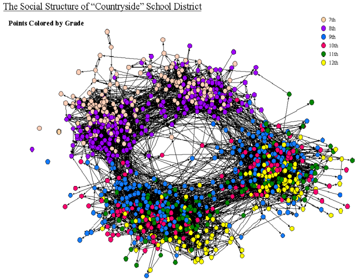
Only small or sparse networks can be displayed readably.

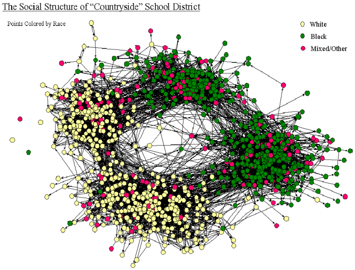
On large networks graph drawing algorithms can reveal their overall structure.

Can we explain the obtained structure?

Visualization: initial network exploration, reporting results, story telling.

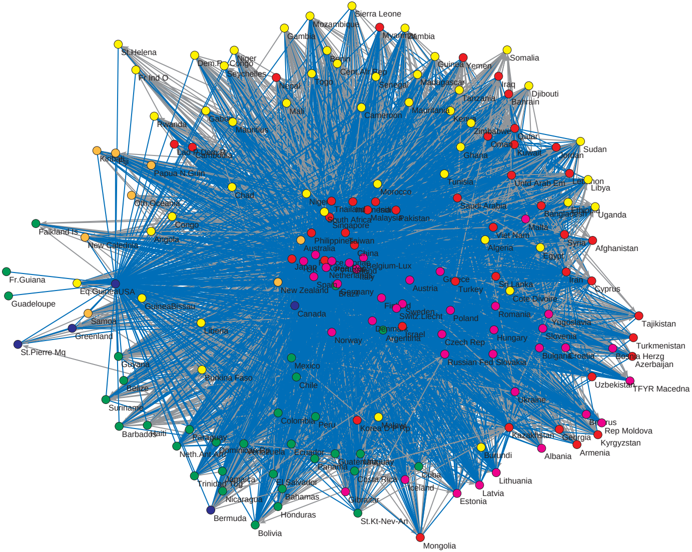
Display of properties – school (Moody)



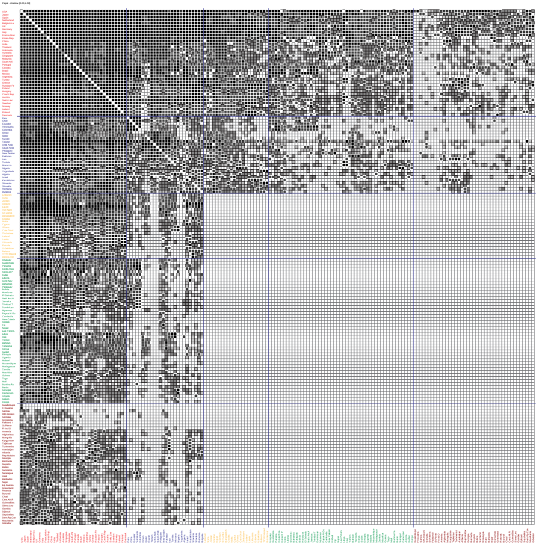


Besides the graph, we need data to understand the network!

A display of World Trade 1999 network



Matrix display of World Trade 1999 network



World trade 1999 clustering

Feenstra, RC, Lipsey, RE, Deng, H, Ma, AC, Mo, H: World Trade Flows: 1962-2000; info → weights in 1000 USD.

```
read World Trade 1999 Pajek project file
button Info network
Network/Info/Line values
Network/Create new network/Transform/Line values/Ln
Network/Create Vector/Centrality/Weighted degree/All
Vector/Make permutation
Permutation/Mirror permutation
File/Network/Export as matrix/EPS/Using permutation

Cluster/Create complete cluster
Operations/Network+cluster/Dissimilarity*/Network based/
   d5 [1]
manually reorder and close hierarchy
Hierarchy/Make partition
Hierarchy/Make permutation
select ln network as the First
File/Network/Export as matrix/EPS/Using
   permutation+partition
```

Types of networks

Besides ordinary (directed, undirected, mixed) networks some extended types of networks are also used:

- *2-mode networks*, bipartite (valued) graphs – networks between two disjoint sets of nodes $\mathbf{N} = ((\mathcal{U}, \mathcal{V}), \mathcal{L}, \mathcal{P}, \mathcal{W})$
- *multi-relational networks* $\mathbf{N} = (\mathcal{V}, (\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k), \mathcal{P}, \mathcal{W})$
- *temporal networks*, dynamic graphs – networks changing over time $\mathbf{N}_T = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W}, T)$
- specialized networks: representation of genealogies as *p-graphs*; *Petri's nets*, ...

The network (input) file formats should provide means to express all these types of networks. All interesting data should be recorded (respecting privacy).

Pictures in SVG: ***66 days***.

Multi-relational temporal network – KEDS/WEIS

```
% Recoded by WEISmonths, Sun Nov 28 21:57:00 2004
% from http://www.ku.edu/~keds/data.dir/balk.html
*vertices 325
1 "AFG" [1-*]
2 "AFR" [1-*]
3 "ALB" [1-*]
4 "ALBMED" [1-*]
5 "ALG" [1-*]
 ...
318 "YUGGOV" [1-*]
319 "YUGMAC" [1-*]
320 "YUGMED" [1-*]
321 "YUGMTN" [1-*]
322 "YUGSER" [1-*]
323 "ZAI" [1-*]
324 "ZAM" [1-*]
325 "ZIM" [1-*]
*arcs :0 "*** ABANDONED"
*arcs :10 "YIELD"
*arcs :11 "SURRENDER"
*arcs :12 "RETREAT"
 ...
*arcs :223 "MIL ENGAGEMENT"
*arcs :224 "RIOT"
*arcs :225 "ASSASSINATE TORTURE"
*arcs
224: 314 153 1 [4]           890402 YUG    KSV    224  (RIOT) RIOT-TORN
212: 314 83 1 [4]            890404 YUG    ETHALB 212  (ARREST PERSON) ALB ETHNI
224: 3 83 1 [4]              890407 ALB    ETHALB 224  (RIOT) RIOTS
123: 83 153 1 [4]            890408 ETHALB KSV    123  (INVESTIGATE) PROBING
 ...
42: 105 63 1 [175]           030731 GER    CYP    042  (ENDORSE) GAVE SUPPORT
212: 295 35 1 [175]          030731 UNWCT  BOSSER 212  (ARREST PERSON) SENTENCED
43: 306 87 1 [175]           030731 VAT    EUR    043  (RALLY) RALLIED
13: 295 35 1 [175]           030731 UNWCT  BOSSER 013  (RETRACT) CLEARED
121: 295 22 1 [175]          030731 UNWCT  BAL    121  (CRITICIZE) CHARGES
122: 246 295 1 [175]         030731 SER    UNWCT  122  (DENIGRATE) TESTIFIED
121: 35 295 1 [175]          030731 BOSSER UNWCT  121  (CRITICIZE) ACCUSED
```

Kansas Event Data System *KEDS*

Important nodes and links in network

To identify important / interesting elements (nodes, links) in a network we often try to express our intuition about important / interesting elements using an appropriate measure (index, weight) following the scheme

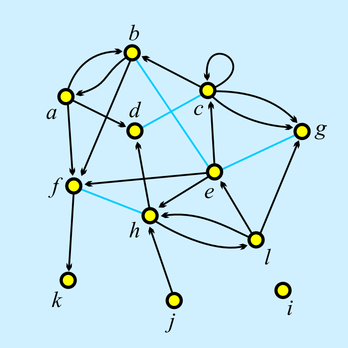
> *larger is the measured value of an element,*
> *more important / interesting is this element*

Too often, in the analysis of networks, researchers uncritically pick some measures from the literature. For a formal approach see Roberts.

The (importance) measure can be obtained as input data – an *observed* property, or computed from the network description – a *structural* property.

An interesting question is studying associations among structural and observed properties. Can an observed property be explained with some structural property/ies?

Degrees



degree of node v, $\deg(v)$ = number of links with v as an endnode;

indegree of node v, $\mathrm{indeg}(v)$ = number of links with v as a terminal node (endnode is both initial and terminal);

outdegree of node v, $\mathrm{outdeg}(v)$ = number of links with v as an initial node.

initial node $v \Leftrightarrow \mathrm{indeg}(v) = 0$

terminal node $v \Leftrightarrow \mathrm{outdeg}(v) = 0$

$$n = 12,\ m = 23,\ \mathrm{indeg}(e) = 3,\ \mathrm{outdeg}(e) = 5,\ \deg(e) = 6$$

$$\sum_{v \in \mathcal{V}} \mathrm{indeg}(v) = \sum_{v \in \mathcal{V}} \mathrm{outdeg}(v) = |\mathcal{A}| + 2|\mathcal{E}| - |\mathcal{E}_0|, \ \sum_{v \in \mathcal{V}} \deg(v) = 2|\mathcal{L}| - |\mathcal{L}_0|$$

Node indices

The most important distinction between different node *indices* is based on the view/decision of whether the network is considered directed or undirected.

Closeness, betweenness, clustering, eigen-vector, ...

Weighted degree, indegree and outdegree

$$\text{wdeg}(v) = \sum_{e \in \text{star}(v)} w(e),$$

$$\text{windeg}(v) = \sum_{e \in \text{instar}(v)} w(e) \quad \text{and} \quad \text{woutdeg}(v) = \sum_{e \in \text{outstar}(v)} w(e)$$

Hubs and authorities

To each node v of a network $\mathbf{N} = (\mathcal{V}, \mathcal{L})$ we assign two values: quality of its content (*authority*) x_v and quality of its references (*hub*) y_v.

A good authority is selected by good hubs; and good hub points to good authorities (see Kleinberg).

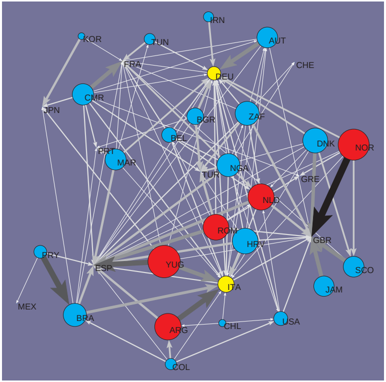
$$x_v = \sum_{u:(u,v)\in\mathcal{L}} y_u \qquad \text{and} \qquad y_v = \sum_{u:(v,u)\in\mathcal{L}} x_u$$

Let $\mathbf{W}$ be a matrix of network $\mathbf{N}$ and $\mathbf{x}$ and $\mathbf{y}$ authority and hub vectors. Then we can write these two relations as $\mathbf{x} = \mathbf{W}^T\mathbf{y}$ and $\mathbf{y} = \mathbf{W}\mathbf{x}$.

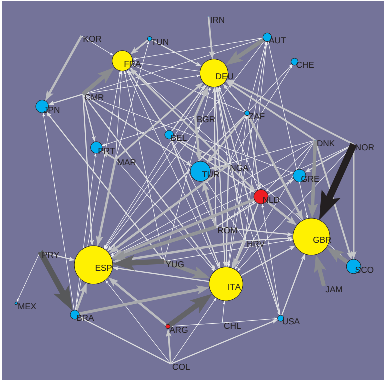
We start with $\mathbf{y} = [1, 1, \ldots, 1]$ and then compute new vectors $\mathbf{x}$ and $\mathbf{y}$. After each step we normalize both vectors. We repeat this until they stabilize.

We can show that this procedure converges. The limit vector $\mathbf{x}^*$ is the principal eigen vector of matrix $\mathbf{W}^T\mathbf{W}$; and $\mathbf{y}^*$ of matrix $\mathbf{W}\mathbf{W}^T$.

. . . Hubs and authorities: football



Exporters (hubs)



Importers (authorities)

Example: Krebs, Krempl. World Cup 1998 in Paris, 22 national teams. A player from first country is playing in the second country.

Important links – Weighted networks

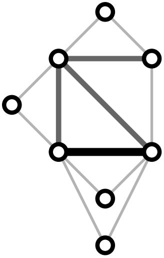
A very important characteristic of weight is its nature – should it be considered a *similarity* (larger is the link weight more similar are its end-nodes) or a *dissimilarity* (larger is the link weight more different are its end-nodes; weight for links with equal end-nodes is usually 0)?

Structural weights are computed based on network structure. For example

$$w(e(u,v)) = \deg(u) \cdot \deg(v)$$

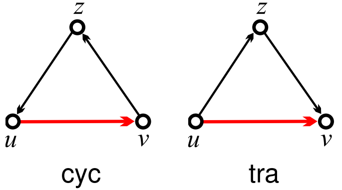
More substantial weights exist.

Triangular network



Let **G** be a simple undirected graph. A *triangular network* $\mathbf{N}_T(\mathbf{G}) = (\mathcal{V}, \mathcal{E}_T, w)$ determined by **G** is a subgraph $\mathbf{G}_T = (\mathcal{V}, \mathcal{E}_T)$ of **G** which set of edges $\mathcal{E}_T$ consists of all triangular edges of $\mathcal{E}(\mathbf{G})$. For $e \in \mathcal{E}_T$ the weight $w(e)$ equals to the number of different triangles in **G** to which e belongs. Triangular networks can be used to efficiently identify dense clique-like parts of a graph. If an edge e belongs to a k-clique in **G** then $w(e) \geq k - 2$.

If a graph **G** is mixed we replace edges with pairs of opposite arcs. Let $\mathbf{G} = (\mathcal{V}, \mathcal{A})$ be a simple directed graph without loops. For a selected arc $(u, v) \in \mathcal{A}$ there are only two different types of directed triangles: **cyc**lic and **tra**nsitive.



Multiplication of networks

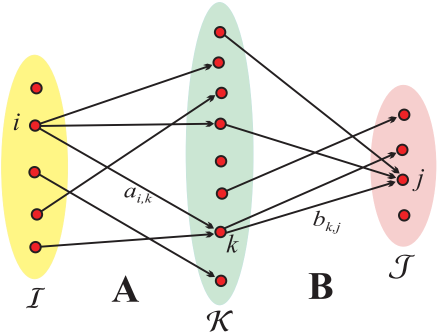
To a simple (no parallel arcs) two-mode *network* $\mathbf{N} = (\mathcal{I}, \mathcal{J}, \mathcal{A}, w)$; where $\mathcal{I}$ and $\mathcal{J}$ are sets of *nodes*, $\mathcal{A}$ is a set of *arcs* linking $\mathcal{I}$ and $\mathcal{J}$, and $w : \mathcal{A} \to \mathbb{R}$ (or some other semiring) is a *weight*; we can assign a *network matrix* $\mathbf{W} = [w_{i,j}]$ with elements: $w_{i,j} = w(i,j)$ for $(i,j) \in \mathcal{A}$ and $w_{i,j} = 0$ otherwise.

Given a pair of compatible networks $\mathbf{N}_A = (\mathcal{I}, \mathcal{K}, \mathcal{A}_A, w_A)$ and $\mathbf{N}_B = (\mathcal{K}, \mathcal{J}, \mathcal{A}_B, w_B)$ with corresponding matrices $\mathbf{A}_{\mathcal{I}\times\mathcal{K}}$ and $\mathbf{B}_{\mathcal{K}\times\mathcal{J}}$ we call a *product of networks* $\mathbf{N}_A$ and $\mathbf{N}_B$ a network $\mathbf{N}_C = (\mathcal{I}, \mathcal{J}, \mathcal{A}_C, w_C)$, where $\mathcal{A}_C = \{(i,j) : i \in \mathcal{I}, j \in \mathcal{J}, c_{i,j} \neq 0\}$ and $w_C(i,j) = c_{i,j}$ for $(i,j) \in \mathcal{A}_C$. The product matrix $\mathbf{C} = [c_{i,j}]_{\mathcal{I}\times\mathcal{J}} = \mathbf{A} * \mathbf{B}$ is defined in the standard way

$$c_{i,j} = \sum_{k\in\mathcal{K}} a_{i,k} \cdot b_{k,j}$$

In the case when $\mathcal{I} = \mathcal{K} = \mathcal{J}$ we are dealing with ordinary one-mode networks (with square matrices).

Multiplication of networks



$$c_{i,j} = \sum_{k \in N_A(i) \cap N_B^-(j)} a_{i,k} \cdot b_{k,j}$$

If all weights in networks $\mathbf{N}_A$ and $\mathbf{N}_B$ are equal to 1 the value of $c_{i,j}$ counts the number of ways we can go from $i \in \mathcal{I}$ to $j \in \mathcal{J}$ passing through $\mathcal{K}$, $c_{i,j} = |N_A(i) \cap N_B^-(j)|$.

Multiplication of networks

The standard matrix multiplication has the complexity $O(|\mathcal{I}| \cdot |\mathcal{K}| \cdot |\mathcal{J}|)$ – it is too slow to be used for large networks. For sparse large networks we can multiply much faster considering only nonzero elements.

for k **in** $\mathcal{K}$ **do**
  for (i, j) **in** $N_A^-(k) \times N_B(k)$ **do**
    if $\exists c_{i,j}$ **then** $c_{i,j} := c_{i,j} + a_{i,k} \cdot b_{k,j}$
    else new $c_{i,j} := a_{i,k} \cdot b_{k,j}$

`Networks/Multiply Networks`

In general the multiplication of large sparse networks is a 'dangerous' operation since the result can 'explode' – it is not sparse.

Multiplication of networks

From the network multiplication algorithm we see that each intermediate node $k \in \mathcal{K}$ adds to a product network a complete two-mode subgraph $K_{N_A^-(k),N_B(k)}$ (or, in the case $\mathcal{I} = \mathcal{J}$, a complete subgraph $K_{N(k)}$). If both degrees $\deg_A(k) = |N_A^-(k)|$ and $\deg_B(k) = |N_B(k)|$ are large then already the computation of this complete subgraph has a quadratic (time and space) complexity – the result 'explodes'.

If at least one of the sparse networks $\mathbf{N}_A$ and $\mathbf{N}_B$ has small maximal degree on $\mathcal{K}$ then also the resulting product network $\mathbf{N}_C$ is sparse.

If for the sparse networks $\mathbf{N}_A$ and $\mathbf{N}_B$ there are in $\mathcal{K}$ only few nodes with large degree and no one among them with large degree in both networks then also the resulting product network $\mathbf{N}_C$ is sparse.

Projections of two-mode networks

Often we transform a two-mode network $\mathbf{N} = (\mathcal{U}, \mathcal{V}, \mathcal{E}, w)$ into an ordinary (one-mode) network $\mathbf{N}_1 = (\mathcal{U}, \mathcal{E}_1, w_1)$ or/and $\mathbf{N}_2 = (\mathcal{V}, \mathcal{E}_2, w_2)$, where $\mathcal{E}_1$ and w_1 are determined by the matrix $\mathbf{W}^{(1)} = \mathbf{W}\mathbf{W}^T$,

$$w^{(1)}_{uv} = \sum_{z \in \mathcal{V}} w_{uz} \cdot w^T_{zv} = \sum_{z \in \mathcal{V}} w_{uz} \cdot w_{vz}.$$

Evidently $w^{(1)}_{uv} = w^{(1)}_{vu}$. There is an edge $(u : v) \in \mathcal{E}_1$ in $\mathbf{N}_1$ iff $N(u) \cap N(v) \neq \emptyset$. Its weight is $w_1(u, v) = w^{(1)}_{uv}$.

The network $\mathbf{N}_2$ is determined in a similar way by the matrix $\mathbf{W}^{(2)} = \mathbf{W}^T\mathbf{W}$.

The networks $\mathbf{N}_1$ and $\mathbf{N}_2$ are analyzed using standard methods.

`Network/2-Mode Network/2-Mode to 1-Mode/Rows`

Derived networks

WA – works $\times$ authors – authorship network
WK – works $\times$ keywords
Ci – works $\times$ works – citation network

$\mathbf{AK} = \mathbf{WA}^T * \mathbf{WK}$ – authors $\times$ keywords

$\mathbf{Co} = \mathbf{WA}^T * \mathbf{WA}$ – coauthorship

$\mathbf{ACi} = \mathbf{WA}^T * \mathbf{Ci} * \mathbf{WA}$ – citations between authors

co_{ij} = the number of works that authors i and j wrote together

It holds: $co_{ij} = co_{ji}$. Using the weights co_{ij} we can determine the Salton's cosine similarity or Ochiai coefficient between authors i and j as

$$S(i,j) = \cos(i,j) = \frac{co_{ij}}{\sqrt{co_{ii}co_{jj}}}$$

Properties of Salton index

The Salton index has the following properties

1. $S(i,j) \in [-1,1]$
2. $S(i,j) = S(j,i)$
3. $S(i,i) = 1$
4. $wa_{pi} \in \mathbb{R}_0^+ \Rightarrow S(u,t) \in [0,1]$
5. $S(\alpha i, \beta j) = S(i,j), \quad \alpha, \beta > 0$
6. $S(\alpha i, i) = 1, \quad \alpha > 0$

Normalizations

In networks obtained from large two-mode networks, there are often huge differences in weights. Therefore it is not interesting to compare the nodes according to the raw data – the nodes with large weights will prevail. First, we have to *normalize* the network to make the weights comparable.

There exist several ways how to do this. Some of them are presented in the following. They can be used also on other weighted networks. Often a given weighted network is essentially a projection of a two-mode network; sometimes without loops. Assume that the network is described with a square matrix **C**.

Let us denote the row sum $R(u) = \sum_{v \in \mathcal{V}} c[u, v]$, which is the total of weights from node u to other nodes. Let $C(v) = \sum_{u \in \mathcal{V}} c[u, v]$ denote the column sum for the node v. If the network is a projection of a two-mode network we set $R(u) = C(u) = c[u, u]$.

. . . Normalizations

Markov (also known as stochastic, affinity, output, row) normalization: For $R(u) > 0$

$$c_M[u,v] = \frac{c[u,v]}{R(u)}$$

If $R(u) = 0$ also $c_M[u,v] = 0$.

Jaccard-like normalization:

$$c_J[u,v] = \frac{c[u,v]}{R(u) + R(v) - c[u,v]}$$

Salton-like normalization:

$$c_S[u,v] = \frac{c[u,v]}{\sqrt{R(u) \cdot R(v)}}$$

. . . Normalizations

Balassa or activity normalization: Let $T = \sum_{u,v} c[u, v]$ be the total sum of weights in the network.

$$A[u, v] = \frac{c[u, v] \cdot T}{R(u) \cdot C(v)}$$

If $A[u, v] > 1$ the measured weight is larger than expected.

$$c_B[u, v] = \log_2 A[u, v] \quad \text{for } A[u, v] > 0$$

. . . Normalizations

$$c_{\min}[u,v] = \frac{c[u,v]}{\min(R(u),R(v))} \qquad c_{\max}[u,v] = \frac{c_{uv}}{\max(c_{uu},c_{vv})}$$

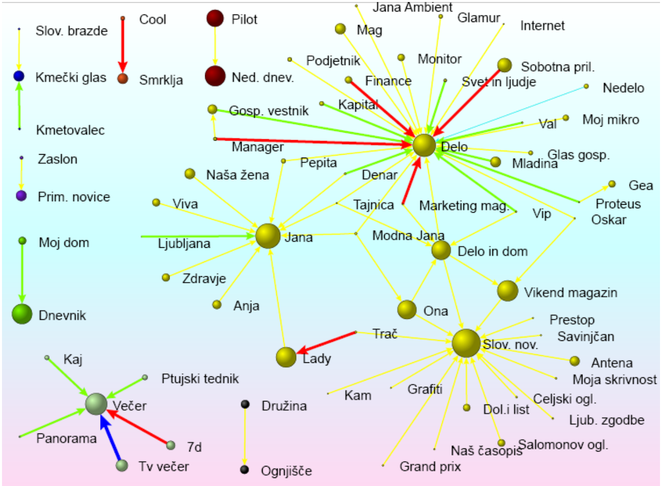
$$c_{\mathrm{minDir}}[u,v] = \begin{cases} \dfrac{c[u,v]}{R(u)} & R(u) \leq R(v) \\ 0 & \textit{otherwise} \end{cases}$$

$$c_{\mathrm{maxDir}}[u,v] = \begin{cases} \dfrac{c[u,v]}{R(v)} & R(u) \leq R(v) \\ 0 & \textit{otherwise} \end{cases}$$

After a selected normalization the important parts of the network are obtained by link-cuts or islands approaches.

Reuters Terror News: GeoDeg, MaxDir, MinDir. Slovenian journals.

minDir of Slovenian journals 2000



Over 100000 people were asked in the years 1999 and 2000 about the journals they read.
They mentioned 124 different journals. (source Cati)

Normalized co-authorship network

Let $\mathbf{N}$ be the normalized version $\forall p \in W : \sum_{i\in A} n_{pi} \in \{0,1\}$ obtained from $\mathbf{WA}$ by $n_{pi} = wa_{pi} / \max(1, \mathrm{outdeg}_{WA}(p))$, or by some other rule determining the author's contribution – the *fractional* approach. Then the *normalized co-authorship network* is

$$\mathbf{Cn} = \mathbf{N}^T \cdot \mathbf{N}$$

cn_{ij} = the total contribution of 'collaboration' of authors i and j to works.

It holds $cn_{ij} = cn_{ji}$ and $\sum_{i\in A}\sum_{j\in A} n_{pi} n_{pj} = 1$.

The total contribution of a complete subgraph corresponding to the authors of a work p is 1.

$$\sum_{i\in A}\sum_{j\in A} cn_{ij} = |W|$$

Normalized strict co-authorship network

Newman defined a *strict normalization* $\mathbf{N}'$ obtained from **WA** by $n'_{pi} = wa_{pi} / \max(1, \mathrm{outdeg}_{WA}(p) - 1)$. Then the *normalized strict co-authorship network* is

$$\mathbf{Cn}' = \mathbf{N}^T \cdot \mathbf{N}'$$

The diagonal (loops) of the so-obtained network $\mathbf{Cn}'$ is set to 0.

The network $\mathbf{Cn}'$ doesn't consider the contribution of single-author works.

Size of network

The size of a network/graph is expressed by two numbers: number of nodes $n = |\mathcal{V}|$ and number of links $m = |\mathcal{L}|$.

In a *simple undirected* graph (no parallel edges, no loops) $m \leq \frac{1}{2}n(n-1)$; and in a *simple directed* graph (no parallel arcs) $m \leq n^2$.

Small networks (some tens of nodes) – can be represented by a picture and analyzed by many algorithms (***UCINET***, ***NetMiner***).

Also, *middle size* networks (some hundreds of nodes) can still be represented by a picture (!?), but some analytical procedures can't be used.

Till 1990 most networks were small – they were collected by researchers using surveys, observations, archival records, . . . The advances in IT allowed to create networks from the data already available in the computer(s). *Large* networks became reality. Large networks are too big to be displayed in detail; special

Large networks

Large network – several thousands or millions of nodes. Can be stored in the computer's memory – otherwise *huge* network. 64-bit computers!

Jure Leskovec: SNAP – Stanford Large Network Dataset Collection

Social networks

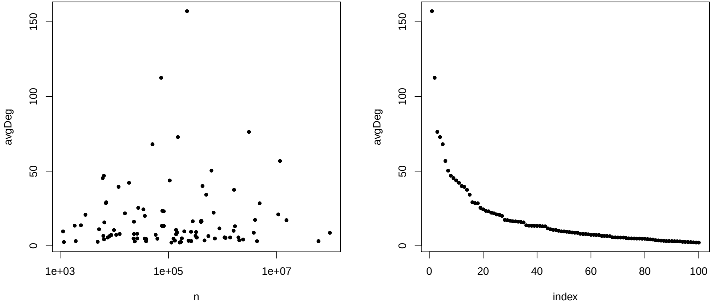
Name	Type	Nodes	Edges	Description
ego-Facebook	Undirected	4,039	88,234	Social circles from Facebook (anonymized)
ego-Gplus	Directed	107,614	13,673,453	Social circles from Google+
ego-Twitter	Directed	81,306	1,768,149	Social circles from Twitter
soc-Epinions1	Directed	75,879	508,837	Who-trusts-whom network of Epinions.com
soc-LiveJournal1	Directed	4,847,571	68,993,773	LiveJournal online social network
soc-Pokec	Directed	1,632,803	30,622,564	Pokec online social network
soc-Slashdot0811	Directed	77,360	905,468	Slashdot social network from November 2008
soc-Slashdot0922	Directed	82,168	948,464	Slashdot social network from February 2009
wiki-Vote	Directed	7,115	103,689	Wikipedia who-votes-on-whom network

Networks with ground-truth communities

Name	Type	Nodes	Edges	Communities	Description
com-LiveJournal	Undirected, Communities	3,997,962	34,681,189	287,512	LiveJournal online social network
com-Friendster	Undirected, Communities	65,608,366	1,806,067,135	957,154	Friendster online social network
com-Orkut	Undirected, Communities	3,072,441	117,185,083	6,288,363	Orkut online social network

Dunbar's number

Average degrees of the SNAP and Konect networks



Average degree $\bar{d} = \frac{1}{n}\sum_{v\in V} \deg(v) = \frac{2m}{n}$. Most real-life large networks are *sparse* – the number of nodes and links are of the same order. This property is also known as a Dunbar's number.

The basic idea is that if each node has to spend for each link a certain amount of "energy" to maintain the links to selected other nodes then since it has a limited "energy" at its disposal, the number of links should be limited. In human networks, Dunbar's number is between 100 and 150.

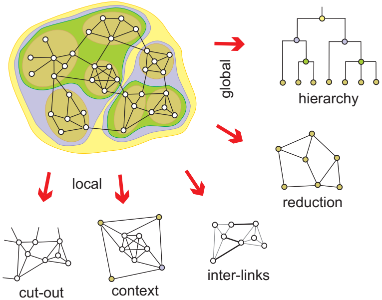
Complexity of algorithms

Let us look to time complexities of some typical algorithms:

	$T(n)$	1.000	10.000	100.000	1.000.000	10.000.000
LinAlg	$O(n)$	0.00 s	0.015 s	0.17 s	2.22 s	22.2 s
LogAlg	$O(n \log n)$	0.00 s	0.06 s	0.98 s	14.4 s	2.8 m
SqrtAlg	$O(n\sqrt{n})$	0.01 s	0.32 s	10.0 s	5.27 m	2.78 h
SqrAlg	$O(n^2)$	0.07 s	7.50 s	12.5 m	20.8 h	86.8 d
CubAlg	$O(n^3)$	0.10 s	1.67 m	1.16 d	3.17 y	3.17 ky

For the interactive use on large graphs already quadratic algorithms, $O(n^2)$, are too slow.

Approaches to large networks



In the analysis of a *large* network (several thousands or millions of nodes, the network can be stored in computer memory) we can't display it in its totality; also there are only a few algorithms available.

To analyze a large network we can use a statistical approach or we can identify smaller (sub) networks that can be analyzed further using more sophisticated methods.

Measured properties – Input data

- numeric → **`vector`**
- ordinal → **`permutation`**
- nominal → **`clustering`** (partition)

Computed properties

global: number of nodes, edges/arcs, components; maximum core number, . . .

local: degrees, cores, indices (betweeness, hubs, authorities, . . .)

inspections: partition, vector, values of lines, . . .

Associations between computed (structural) data and input (measured) data.

. . . Statistics

The global computed properties are reported by **Pajek**'s commands or can be seen using the **Info** option. In *repetitive* commands they are stored in vectors.

The local properties are computed by **Pajek**'s commands and stored in vectors or partitions. To get information about their distribution use the **Info** option.

As an example, let us look at The Edinburgh Associative Thesaurus network. The EAT is a network of word associations collected from subjects (students). The weight on the arcs is the count of word associations.

```
File/Network/Read  EATnewSR.net
Info/Network/General
```

It has 23219 nodes and 325593 arcs (564 loops); number of links with value=1 is 227459; and Average Degree = 28.045 .

. . . Statistics

To identify the nodes with the largest degree:

```
Network/Create vector/Centrality/Degree/All
button Info/Vector  +10
```

The largest degrees have the nodes:

	vertex	deg	label
1	12720	1108	ME
2	12459	1074	MAN
3	8878	878	GOOD
4	18122	875	SEX
5	13793	803	NO
6	13181	799	MONEY
7	23136	732	YES
8	15080	723	PEOPLE
9	13948	720	NOTHING
10	22973	716	WORK

What about indegree and outdegree?

Distribution of weights

```
Network/Info/Line values
```

. . . weighted indegrees

To identify the words selected by the largest number of respondents we look to their weighted indegrees

```
Network/Create vector/Centrality/Weighted degree/In
button Info/Vector  +20
```

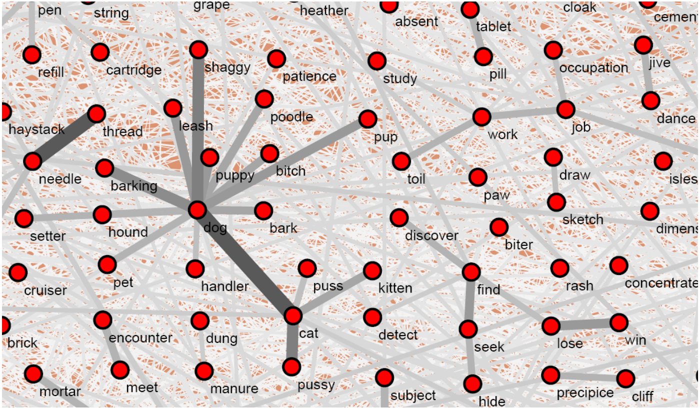
The result is:

	wideg	word	wideg	word	
1	4387	MONEY	11	2019	TREE
2	3299	WATER	12	1988	GOOD
3	2918	FOOD	13	1972	HOUSE
4	2515	ME	14	1896	BIRD
5	2435	MAN	15	1891	UP
6	2434	CAR	16	1881	CHURCH
7	2224	SEA	17	1802	TIME
8	2154	SEX	18	1795	FIRE
9	2100	HORSE	19	1762	SHIP
10	2073	DOG	20	1722	MUSIC

Big picture of EAT at Ars Electronica 2004

Part of the big picture of EAT

The glasses in this case are based on ordering the edges in increasing order of their values and drawing them in this order – stronger edges cover the weaker ones. The picture emphasizes the strongest substructures; the remaining elements form a background.



Statistics with **Pajek** and R

Pajek supports interaction with statistical program R and the use of other external programs as tools (menu `Tools`).

In **Pajek** we determine the degrees of nodes and submit them to R

```
Network/Create Vector/Centrality/Degree/All
Tools/R/Send to R/Current Vector
```

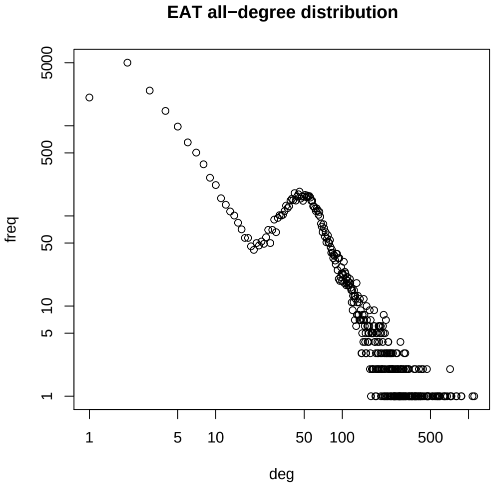
In R we determine their distribution and plot it

```
summary(v2)
t <- table(v2)
x<-as.numeric(names(t))
plot(x,t,log='xy',main='degree distribution',
  xlab='deg',ylab='freq')
```

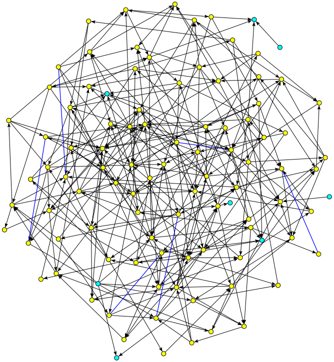
The obtained picture can be saved with `File/Save as` in selected format (PDF or PS for LaTeX; Windows metafile format for inclusion in Word; SVG for WWW).

Attention! The nodes of degree 0 make problems with `log='xy'`.

EAT all-degree distribution



Erdős and Renyi's random graphs



Erdős and Renyi defined a *random graph* as follows: every possible link is included in a graph with a given probability p.

In **Pajek** instead of probability p a more intuitive average degree is used

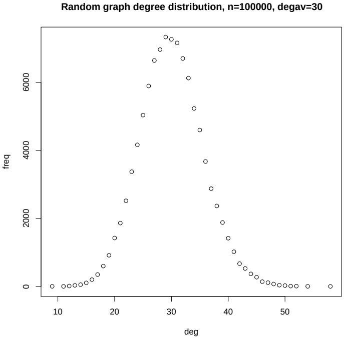
$$\overline{\deg} = \frac{1}{n} \sum_{v \in \mathcal{V}} \deg(v)$$

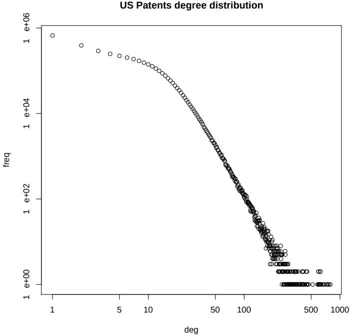
`Network/Create Random Network/Bernoulli/Poisson/Undirected/General [100] [3]`

It holds $p = \frac{m}{m_{max}}$ and, for simple graphs, also $\overline{\deg} = \frac{2m}{n}$.

The random graph in the picture has 100 nodes and an average degree 3.

Degree distribution





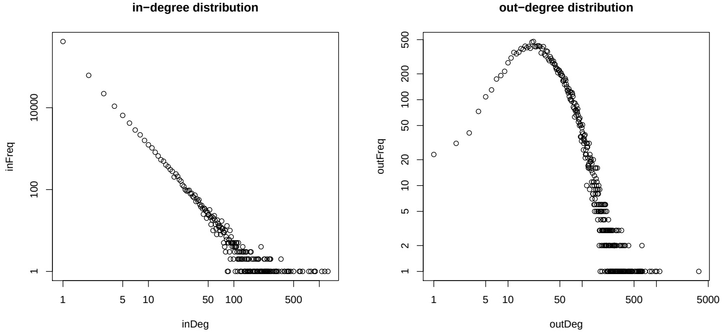
Real-life networks are usually not random in the Erdős/Renyi sense. The analysis of their distributions gave a new view about their structure – Watts (Small worlds), Barabási (nd/networks, Linked).

in/out-degree distributions

We read in **Pajek** the Centrality citation network `cite.net`. First, we remove loops and multiple links. Then we determine the indegrees and outdegrees and call R from **Pajek** submitting all vectors.

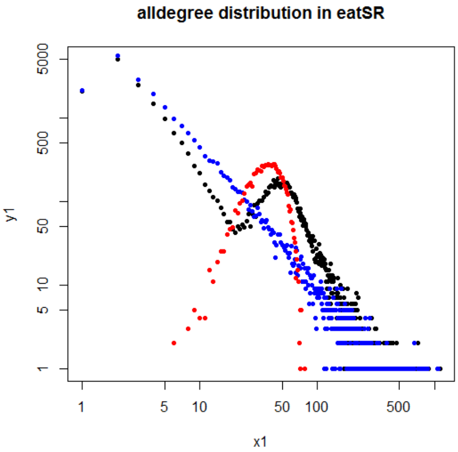
```
#######################################
 R called from Pajek
 The following vectors read:
 v1   :  Input Degree of N1 (548600)
 v2   :  Output Degree of N1 (548600)
 v3   :  All Degree of N1 (548600)
------------------------------------------------------------
> inTab <- table(v1)
> indeg <- as.integer(names(inTab))
> inDeg <- indeg[indeg>0]
> inFreq <- as.vector(inTab[indeg>0])
> plot(inDeg,inFreq,log='xy',main="in-degree distribution")
> ouTab <- table(v2)
> outdeg <- as.integer(names(ouTab))
> outDeg <- outdeg[outdeg>0]
> outFreq <- as.vector(ouTab[outdeg>0])
> plot(outDeg,outFreq,log='xy',main="out-degree distribution")
```

in/out-degree distributions



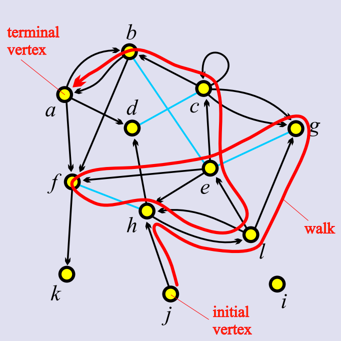
The in-degree distribution is "scale-free"-like – it follows the *power law* $inFreq = k \cdot inDeg^{-\alpha}$. The parameters k and α can be determined using the package of Clauset, Shalizi and Newman. See also Stumpf, et al.: Critical Truths About Power Laws.

EAT all/in/out-degree distributions



It is important to consider the direction of links!

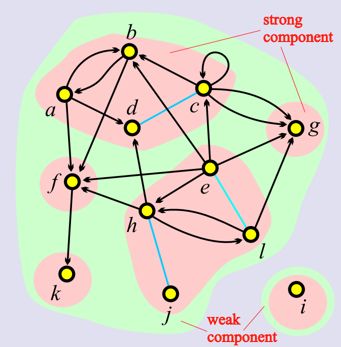
Walks



length $|s|$ of the walk s is the number of links it contains.
$s = (j, h, l, g, e, f, h, l, e, c, b, a)$
$|s| = 11$
A walk is *closed* iff its initial and terminal node coincide.
If we don't consider the direction of the links in the walk we get a *semiwalk* or *chain*.
trail – walk with all links different
path – walk with all nodes different
cycle – closed walk with all internal nodes different

A graph is *acyclic* if it doesn't contain any cycle.

Connectivity



Node u is *reachable* from node v iff there exists a walk with initial node v and terminal node u.

Node v is *weakly connected* with node u iff there exists a semiwalk with v and u as its end-nodes.

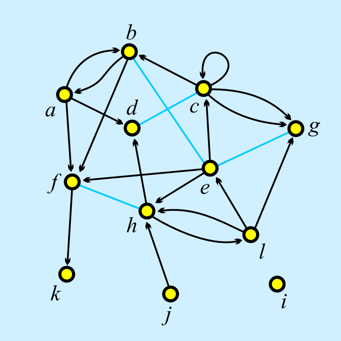
Node v is *strongly connected* with node u iff they are mutually reachable.

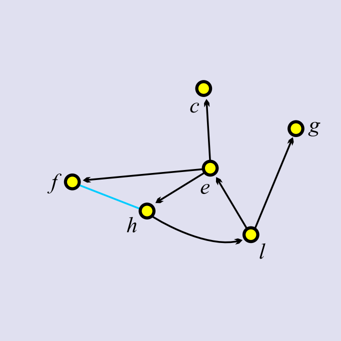
Weak and strong connectivity are equivalence relations.

Equivalence classes induce weak/strong *components*.

`Network/Create Partition/Components/`

Subgraph



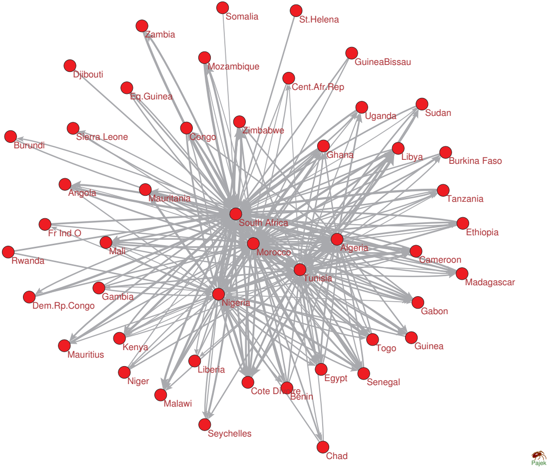


A *subgraph* $\mathbf{H} = (\mathcal{V}', \mathcal{L}')$ of a given graph $\mathbf{G} = (\mathcal{V}, \mathcal{L})$ is a graph which set of links is a subset of set of links of **G**, $\mathcal{L}' \subseteq \mathcal{L}$, its node set is a subset of set of nodes of **G**, $\mathcal{V}' \subseteq \mathcal{V}$, and it contains all endnodes of $\mathcal{L}'$.

A subgraph can be *induced* by a given subset of nodes or links. It is a *spanning* subgraph iff $\mathcal{V}' = \mathcal{V}$.

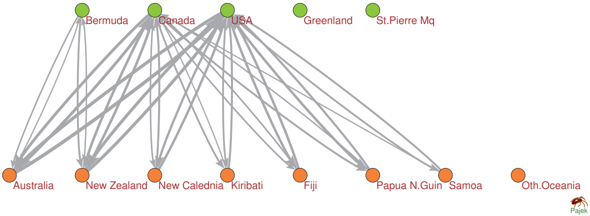
To obtain a *subnetwork* also the properties/weights have to be restricted to $\mathcal{V}'$ and $\mathcal{L}'$).

Cut-out – induced subgraph: World trade Africa




```
select Continet 1999 partition as First
Operations/Network+Partition/Extract/Subnetwork [6]
```

Cut-out: World trade North America : Oceania




```
Operations/Network + Partition/Extract/Subnetwork  [2,7]
Operations/Network + Partition/Transform/Remove lines/
  Inside clusters   [2,7]
```

The nodes can be manually put on a rectangular grid produced by

```
[Draw] Move/Grid
```

Cuts

The standard approach to find interesting groups inside a network is based on properties/weights – they can be *measured* or *computed* from network structure.

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

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

and $\mathcal{L}(\mathcal{V}')$ is the set of links from $\mathcal{L}$ that have both endnodes in $\mathcal{V}'$.

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

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

and $\mathcal{V}(\mathcal{L}')$ is the set of all endnodes of the links from $\mathcal{L}'$.

Cuts in **Pajek**

The threshold value *t* is determined on the basis of distribution of values of weight *w* or property *p*. Usually we are interested in cuts that are not too large, but also not trivial.

Node-cut: *p* stored in a vector

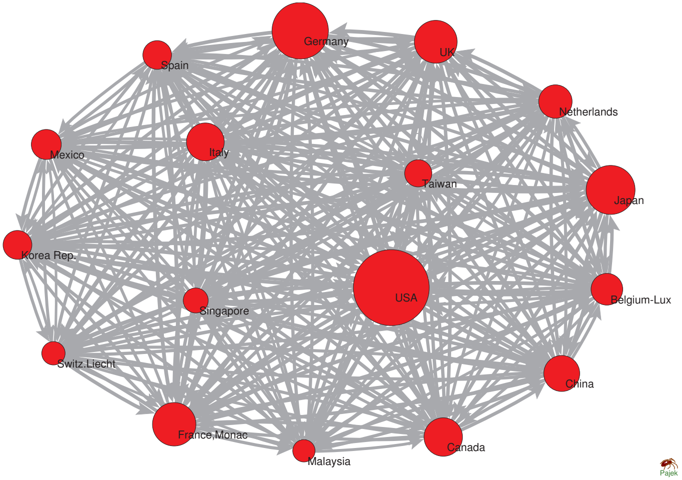
```
Vector/Info  [+30] [#20]
Vector/Make Partition/by Intervals/Selected Thresholds  [t]
Operations/Network + Partition/Extract Subnetwork  [2]
```

Link-cut: weighted network

```
Network/Info/Line values   [#20]
Network/Create New Network/Transform/Remove/Lines with Value/
  lower than   [t]
Network/Create Partition/Degree/All
Operations/Network + Partition/Extract Subnetwork  [1-*]
```

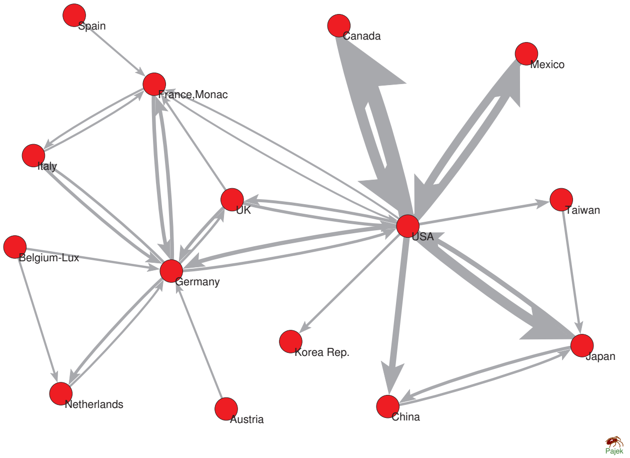
Node-cut: World trade, wdeg=150000

weights devided by 1000



Link-cut: World trade, wdeg=25000

weights devided by 1000



Skeletons

Skeletons are subnetworks in which only the most important links are preserved.

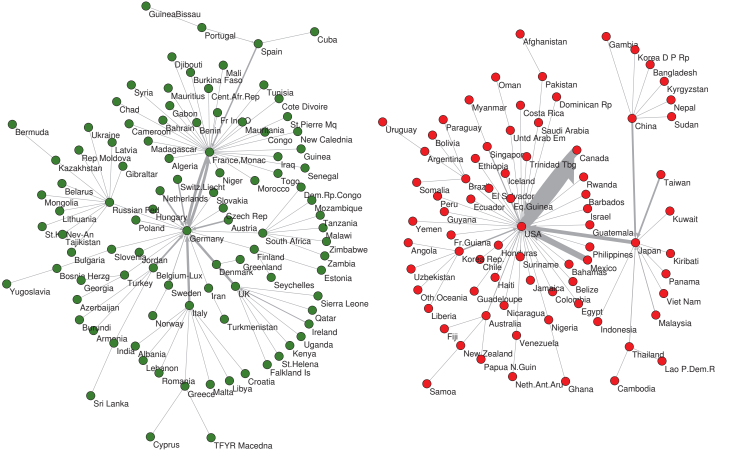
In the *top k-neighbors skeleton* in each node the k links with the highest weights are preserved.

```
Network/Create New Network/Transform/Remove/all arcs from
  each vertex except/ k with highest line values    [k]
```

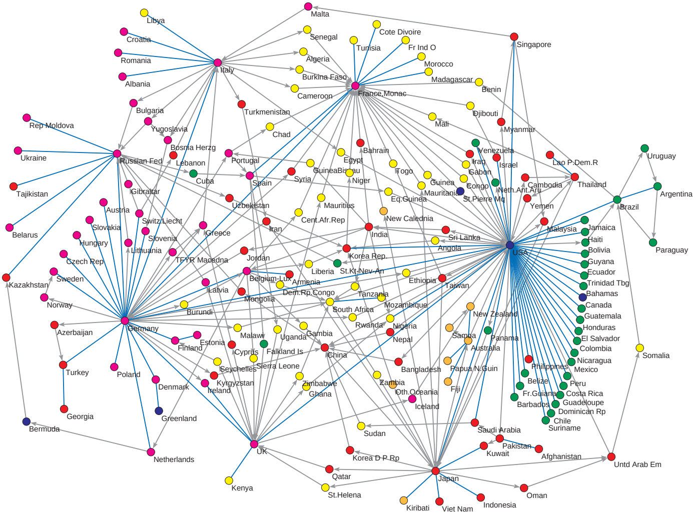
Let the weight w be a dissimilarity ($w' = 1/w$). The Pathfinder algorithm was proposed in eighties by Schvaneveldt [9] for simplification of weighted networks. It removes from the network all links that do not satisfy the triangle inequality – if for a link a shorter path exists connecting its endnodes then the link is removed. The obtained subnetwork is called the *Pathfinder skeleton*.

```
Network/Create New Network/Transform/Line values/Power [-1]
Network/Create New Network/Transform/Reduction/Pathfinder [1]
select the original network as the Second
Networks/Cross-intersection/Second
```

Nearest neighbor skeleton: World trade



Pathfinder skeleton of World Trade 1999 network



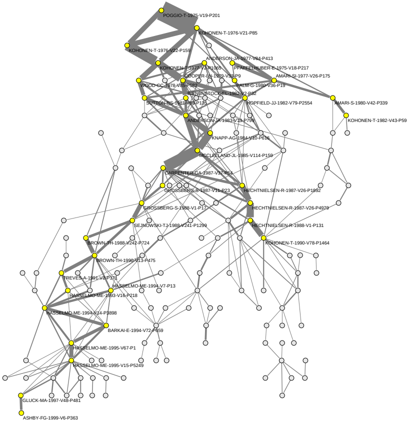
Simple analysis using cuts

We look at the components of $\mathbf{N}(t)$. 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 and not containing important nodes/links 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 node/link-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.

Citation weights

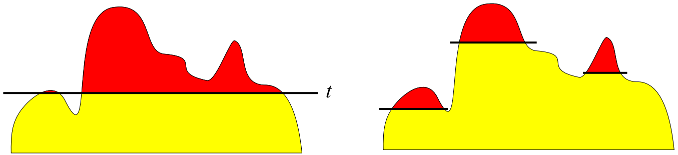


The citation network analysis started in 1964 with the paper of Garfield et al. In 1989 Hummon and Doreian proposed three indices – weights of arcs that are proportional to the number of different source-sink paths passing through the arc. We developed algorithms to efficiently compute these indices.

Main subnetwork (arc-cut at level 0.007) of the SOM (selforganizing maps) citation network (4470 nodes, 12731 arcs). See paper.

Islands

If we represent a given or computed value of nodes / links as a height of nodes / links and we immerse the network into a water up to selected level we get *islands*. Varying the level we get different islands.



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

See details.

. . . Islands

Islands are very general and efficient approach to determine the 'important' subnetworks in a given network.

We have to express the goals of our analysis with a related property of the nodes or weight of the links. Using this property we determine the islands of an appropriate size (in the interval k to K).

In large networks we can get many islands which we have to inspect individually and interpret their content.

An important property of the islands is that they identify locally important subnetworks on different levels. Therefore they detect also emerging groups.

. . . Islands

A set of nodes $C \subseteq \mathcal{V}$ is a *regular node island* in network $\mathbf{N} = (\mathcal{V}, \mathcal{L}, p)$, $p : \mathcal{V} \to \mathbb{R}$ iff it induces a connected subgraph and the nodes from the island are 'higher' than the neighboring nodes

$$\max_{u \in N(C)} p(u) < \min_{v \in c} p(v)$$

A set of nodes $C \subseteq \mathcal{V}$ is a *regular link island* in network $\mathbf{N} = (\mathcal{V}, \mathcal{L}, w)$, $w : \mathcal{L} \to \mathbb{R}$ iff it induces a connected subgraph and the links inside the island are 'stronger related' among them than with the neighboring nodes – in $\mathbf{N}$ there exists a spanning tree $\mathbf{T}$ over C such that

$$\max_{(u,v) \in \mathcal{L}, u \notin C, v \in C} w(u,v) < \min_{(u,v) \in \mathbf{T}} w(u,v)$$

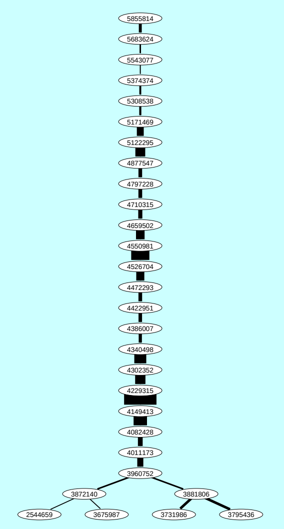
US patents

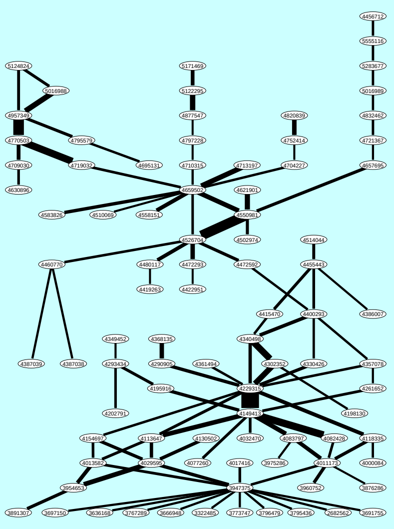
US patents network (Nber, US Patents) has 3 774 768 nodes and16 522 438 arcs (1 loop). Without the loop it is acyclic. The weight of an arc is the proportion of paths through the arc from some initial node to some terminal node – Hummon and Doreian's weight. We determined al (2,90)-islands. The corresponding subnetwork has 470 137 nodes, 307472 arcs and for different k: C_2 =187 610, C_5 =8 859, C_{30} =101, C_{50} =30, . . . islands. Rolex

[1]	0	139793	29670	9288	3966	1827	997	578	362	250
[11]	190	125	104	71	47	37	36	33	21	23
[21]	17	16	8	7	13	10	10	5	5	5
[31]	12	3	7	3	3	3	2	6	6	2
[41]	1	3	4	1	5	2	1	1	1	1
[51]	2	3	3	2	0	0	0	0	0	1
[61]	0	0	0	0	1	0	0	2	0	0
[71]	0	0	1	1	0	0	0	1	0	0
[81]	2	0	0	0	0	1	2	0	0	7

Main path and main island in US Patents

Nber, US Patents; $n = 3774768$, $m = 16522438$





Main island – Liquid crystal display

Table 1: Patents on the liquid-crystal display

patent	date	author(s) and title
2544659	Mar 13, 1951	Dreyer. Dichroic light-polarizing sheet and the like and the formation and use thereof
2682562	Jun 29, 1954	Wender, et al. Reduction of aromatic carbinols
3322485	May 30, 1967	Williams. Electro-optical elements utilazing an organic nematic compound
3636168	Jan 18, 1972	Josephson. Preparation of polynuclear aromatic compounds
3666948	May 30, 1972	Mechlowitz, et al. Liquid crystal termal imaging system having an undisturbed image on a disturbed background
3675987	Jul 11, 1972	Rafuse. Liquid crystal compositions and devices
3691755	Sep 19, 1972	Girard. Clock with digital display
3697150	Oct 10, 1972	Wysochi. Electro-optic systems in which an electrophoretic-like or dipolar material is dispersed throughout a liquid crystal to reduce the turn-off time
3731986	May 8, 1973	Fergason. Display devices utilizing liquid crystal light modulation
3767289	Oct 23, 1973	Aviram, et al. Class of stable trans-stilbene compounds, some displaying nematic mesophases at or near room temperature and others in a range up to 100°C
3773747	Nov 20, 1973	Steinstrasser. Substituted azoxy benzene compounds
3795436	Mar 5, 1974	Boller, et al. Nematogenic material which exhibit the Kerr effect at isotropic temperatures
3796479	Mar 12, 1974	Helfrich, et al. Electro-optical light-modulation cell utilizing a nematogenic material which exhibits the Kerr effect at isotropic temperatures
3872140	Mar 18, 1975	Klanderman, et al. Liquid crystalline compositions and method
3876286	Apr 8, 1975	Deutscher, et al. Use of nematic liquid crystalline substances
3881806	May 6, 1975	Suzuki. Electro-optical display device
3891307	Jun 24, 1975	Tsukamoto, et al. Phase control of the voltages applied to opposite electrodes for a cholesteric to nematic phase transition display
3947375	Mar 30, 1976	Gray, et al. Liquid crystal materials and devices
3954653	May 4, 1976	Yamazaki. Liquid crystal composition having high dielectric anisotropy and display device incorporating same
3960752	Jun 1, 1976	Klanderman, et al. Liquid crystal compositions
3975286	Aug 17, 1976	Oh. Low voltage actuated field effect liquid crystals compositions and method of synthesis
4000084	Dec 28, 1976	Hsieh, et al. Liquid crystal mixtures for electro-optical display devices
4011173	Mar 8, 1977	Steinstrasser. Modified nematic mixtures with positive dielectric anisotropy
4013582	Mar 22, 1977	Gavrilovic. Liquid crystal compounds and electro-optic devices incorporating them
4017416	Apr 12, 1977	Inukai, et al. P-cyanophenyl 4-alkyl-4'-biphenylcarboxylate, method for preparing same and liquid crystal compositions using same
4029595	Jun 14, 1977	Ross, et al. Novel liquid crystal compounds and electro-optic devices incorporating them
4032470	Jun 28, 1977	Bloom, et al. Electro-optic device
4077260	Mar 7, 1978	Gray, et al. Optically active cyano-biphenyl compounds and liquid crystal materials containing them
4082428	Apr 4, 1978	Hsu. Liquid crystal composition and method

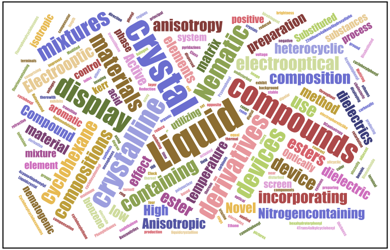
Table 2: Patents on the liquid-crystal display

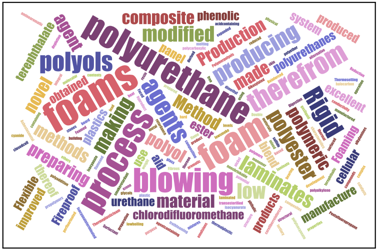
patent	date	author(s) and title
4083797	Apr 11, 1978	Oh. Nematic liquid crystal compositions
4113647	Sep 12, 1978	Coates, et al. Liquid crystalline materials
4118335	Oct 3, 1978	Krause, et al. Liquid crystalline materials of reduced viscosity
4130502	Dec 19, 1978	Eidenschink, et al. Liquid crystalline cyclohexane derivatives
4149413	Apr 17, 1979	Gray, et al. Optically active liquid crystal mixtures and liquid crystal devices containing them
4154697	May 15, 1979	Eidenschink, et al. Liquid crystalline hexahydroterphenyl derivatives
4195916	Apr 1, 1980	Coates, et al. Liquid crystal compounds
4198130	Apr 15, 1980	Boller, et al. Liquid crystal mixtures
4202791	May 13, 1980	Sato, et al. Nematic liquid crystalline materials
4229315	Oct 21, 1980	Krause, et al. Liquid crystalline cyclohexane derivatives
4261652	Apr 14, 1981	Gray, et al. Liquid crystal compounds and materials and devices containing them
4290905	Sep 22, 1981	Kanbe. Ester compound
4293434	Oct 6, 1981	Deutscher, et al. Liquid crystal compounds
4302352	Nov 24, 1981	Eidenschink, et al. Fluorophenylcyclohexanes, the preparation thereof and their use as components of liquid crystal dielectrics
4330426	May 18, 1982	Eidenschink, et al. Cyclohexylbiphenyls, their preparation and use in dielectrics and electrooptical display elements
4340498	Jul 20, 1982	Sugimori. Halogenated ester derivatives
4349452	Sep 14, 1982	Osman, et al. Cyclohexylcyclohexanoates
4357078	Nov 2, 1982	Carr, et al. Liquid crystal compounds containing an alicyclic ring and exhibiting a low dielectric anisotropy and liquid crystal materials and devices incorporating such compounds
4361494	Nov 30, 1982	Osman, et al. Anisotropic cyclohexyl cyclohexylmethyl ethers
4368135	Jan 11, 1983	Osman. Anisotropic compounds with negative or positive DC-anisotropy and low optical anisotropy
4386007	May 31, 1983	Krause, et al. Liquid crystalline naphthalene derivatives
4387038	Jun 7, 1983	Fukui, et al. 4-(Trans-4'-alkylcyclohexyl) benzoic acid 4'''-cyano-4''-biphenylyl esters
4387039	Jun 7, 1983	Sugimori, et al. Trans-4-(trans-4'-alkylcyclohexyl)-cyclohexane carboxylic acid 4'''-cyanobiphenyl ester
4400293	Aug 23, 1983	Romer, et al. Liquid crystalline cyclohexylphenyl derivatives
4415470	Nov 15, 1983	Eidenschink, et al. Liquid crystalline fluorine-containing cyclohexylbiphenyls and dielectrics and electro-optical display elements based thereon
4419263	Dec 6, 1983	Praefcke, et al. Liquid crystalline cyclohexylcarbonitrile derivatives
4422951	Dec 27, 1983	Sugimori, et al. Liquid crystal benzene derivatives
4455443	Jun 19, 1984	Takatsu, et al. Nematic halogen Compound
4456712	Jun 26, 1984	Christie, et al. Bismaleimide triazine composition
4460770	Jul 17, 1984	Petrzilka, et al. Liquid crystal mixture
4472293	Sep 18, 1984	Sugimori, et al. High temperature liquid crystal substances of four rings and liquid crystal compositions containing the same
4472592	Sep 18, 1984	Takatsu, et al. Nematic liquid crystalline compounds
4480117	Oct 30, 1984	Takatsu, et al. Nematic liquid crystalline compounds
4502974	Mar 5, 1985	Sugimori, et al. High temperature liquid-crystalline ester compounds
4510069	Apr 9, 1985	Eidenschink, et al. Cyclohexane derivatives

Table 3: Patents on the liquid-crystal display

patent	date	author(s) and title
4514044	Apr 30, 1985	Gunjima, et al. 1-(Trans-4-alkylcyclohexyl)-2-(trans-4'-(p-substituted phenyl) cyclohexyl)ethane and liquid crystal mixture
4526704	Jul 2, 1985	Petrzilka, et al. Multiring liquid crystal esters
4550981	Nov 5, 1985	Petrzilka, et al. Liquid crystalline esters and mixtures
4558151	Dec 10, 1985	Takatsu, et al. Nematic liquid crystalline compounds
4583826	Apr 22, 1986	Petrzilka, et al. Phenylethanes
4621901	Nov 11, 1986	Petrzilka, et al. Novel liquid crystal mixtures
4630896	Dec 23, 1986	Petrzilka, et al. Benzonitriles
4657695	Apr 14, 1987	Saito, et al. Substituted pyridazines
4659502	Apr 21, 1987	Fearon, et al. Ethane derivatives
4695131	Sep 22, 1987	Balkwill, et al. Disubstituted ethanes and their use in liquid crystal materials and devices
4704227	Nov 3, 1987	Krause, et al. Liquid crystal compounds
4709030	Nov 24, 1987	Petrzilka, et al. Novel liquid crystal mixtures
4710315	Dec 1, 1987	Schad, et al. Anisotropic compounds and liquid crystal mixtures therewith
4713197	Dec 15, 1987	Eidenschink, et al. Nitrogen-containing heterocyclic compounds
4719032	Jan 12, 1988	Wachtler, et al. Cyclohexane derivatives
4721367	Jan 26, 1988	Yoshinaga, et al. Liquid crystal device
4752414	Jun 21, 1988	Eidenschink, et al. Nitrogen-containing heterocyclic compounds
4770503	Sep 13, 1988	Buchecker, et al. Liquid crystalline compounds
4795579	Jan 3, 1989	Vauchier, et al. 2,2'-difluoro-4-alkoxy-4'-hydroxydiphenyls and their derivatives, their production process and their use in liquid crystal display devices
4797228	Jan 10, 1989	Goto, et al. Cyclohexane derivative and liquid crystal composition containing same
4820839	Apr 11, 1989	Krause, et al. Nitrogen-containing heterocyclic esters
4832462	May 23, 1989	Clark, et al. Liquid crystal devices
4877547	Oct 31, 1989	Weber, et al. Liquid crystal display element
4957349	Sep 18, 1990	Clerc, et al. Active matrix screen for the color display of television pictures, control system and process for producing said screen
5016988	May 21, 1991	Iimura. Liquid crystal display device with a birefringent compensator
5016989	May 21, 1991	Okada. Liquid crystal element with improved contrast and brightness
5122295	Jun 16, 1992	Weber, et al. Matrix liquid crystal display
5124824	Jun 23, 1992	Kozaki, et al. Liquid crystal display device comprising a retardation compensation layer having a maximum principal refractive index in the thickness direction
5171469	Dec 15, 1992	Hittich, et al. Liquid-crystal matrix display
5283677	Feb 1, 1994	Sagawa, et al. Liquid crystal display with ground regions between terminal groups
5308538	May 3, 1994	Weber, et al. Supertwist liquid-crystal display
5374374	Dec 20, 1994	Weber, et al. Supertwist liquid-crystal display
5543077	Aug 6, 1996	Rieger, et al. Nematic liquid-crystal composition
5555116	Sep 10, 1996	Ishikawa, et al. Liquid crystal display having adjacent electrode terminals set equal in length
5683624	Nov 4, 1997	Sekiguchi, et al. Liquid crystal composition
5855814	Jan 5, 1999	Matsui, et al. Liquid crystal compositions and liquid crystal display elements

Word clouds for LCD island and foam island





Islands

Dense groups

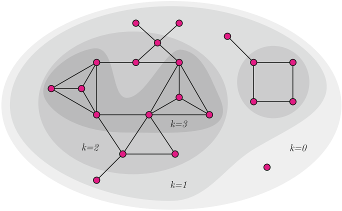
Activities in a part of network usually increase local density (number of nodes and links) in that part of network. Several notions were proposed in attempts to formally describe dense groups in graphs.

Clique of order k is a maximal complete subgraph (isomorphic to K_k), $k \geq 3$.

s-plexes, LS sets, lambda sets, cores, . . .

For all of them, except for cores, it turned out that they are "expensive" to detemine.

Cores and generalized cores



The notion of core was introduced by Seidman in 1983. Let $\mathbf{G} = (\mathcal{V}, \mathcal{E})$ be a graph. A subgraph $\mathbf{H} = (W, \mathcal{E}|W)$ induced by the set W is a k-*core* or a *core of order* k iff $\forall v \in W : \deg_{\mathbf{H}}(v) \geq k$, and $\mathbf{H}$ is a maximal subgraph with this property. The core of maximum order is also called the *main* core.

The *core number* of node v is the highest order of a core that contains this node. The degree $\deg(v)$ can be: in-degree, out-degree, in-degree + out-degree, etc., determining different types of cores.

Properties of cores

From the figure, representing 0, 1, 2 and 3 core, we can see the following properties of cores:

- The cores are nested: $i < j \implies \mathbf{H}_j \subseteq \mathbf{H}_i$
- Cores are not necessarily connected subgraphs.

An efficient algorithm for determining the cores hierarchy is based on the following property:

> *If from a given graph $\mathbf{G} = (\mathcal{V}, \mathcal{E})$ we recursively delete all nodes, and edges incident with them, of degree less than k, the remaining graph is the k-core.*

Generalized cores

The notion of core can be generalized to networks. Let $\mathbf{N} = (\mathcal{V}, \mathcal{E}, w)$ be a network, where $\mathbf{G} = (\mathcal{V}, \mathcal{E})$ is a graph and $w : \mathcal{E} \to \mathbb{R}$ is a function assigning values to edges. A *node property function* on $\mathbf{N}$, or a p-function for short, is a function $p(v, U)$, $v \in \mathcal{V}$, $U \subseteq \mathcal{V}$ with real values. Let $N_U(v) = N(v) \cap U$. Besides degrees and (corrected) clustering coefficient, here are some examples of p-functions:

$$
\begin{aligned}
p_S(v, U) &= \sum_{u \in N_U(v)} w(v, u), \text{ where } w : \mathcal{E} \to \mathbb{R}_0^+ \\
p_M(v, U) &= \max_{u \in N_U(v)} w(v, u), \text{ where } w : \mathcal{E} \to \mathbb{R} \\
p_t(v, \mathcal{U}) &= \frac{|\mathcal{L}(\mathcal{U}) \cap \mathcal{L}(K(N^+(v)))|}{|\mathcal{L}(K(N^+(v)))|} \\
p_k(v, U) &= \text{number of cycles of length } k \text{ through node } v \text{ in } (U, \mathcal{E}|U)
\end{aligned}
$$

The subgraph $\mathbf{H} = (C, \mathcal{E}|C)$ induced by the set $C \subseteq \mathcal{V}$ is a *p-core at level* $t \in \mathbb{R}$ iff $\forall v \in C : t \leq p(v, C)$ and C is a maximal such set.

Additional p-functions

relative density

$$p_\gamma(v,\mathcal{C}) = \frac{\deg(v,\mathcal{C})}{\max_{u\in N(v)} \deg(u)}, \text{ if } \deg(v) > 0;\ 0, \text{ otherwise}$$

diversity

$$p_\delta(v,\mathcal{C}) = \max_{u\in N^+(v,\mathcal{C})} \deg(u) - \min_{u\in N^+(v,\mathcal{C})} \deg(u)$$

average weight

$$p_a(v,\mathcal{C}) = \frac{1}{|N(v,\mathcal{C})|} \sum_{u\in N(v,\mathcal{C})} w(v,u), \text{ if } N(v,\mathcal{C}) \neq \emptyset;\ 0, \text{ otherwise}$$

Generalized cores algorithms

The function p is *monotone* iff it has the property

$$C_1 \subset C_2 \Rightarrow \forall v \in \mathcal{V} : (p(v, C_1) \leq p(v, C_2))$$

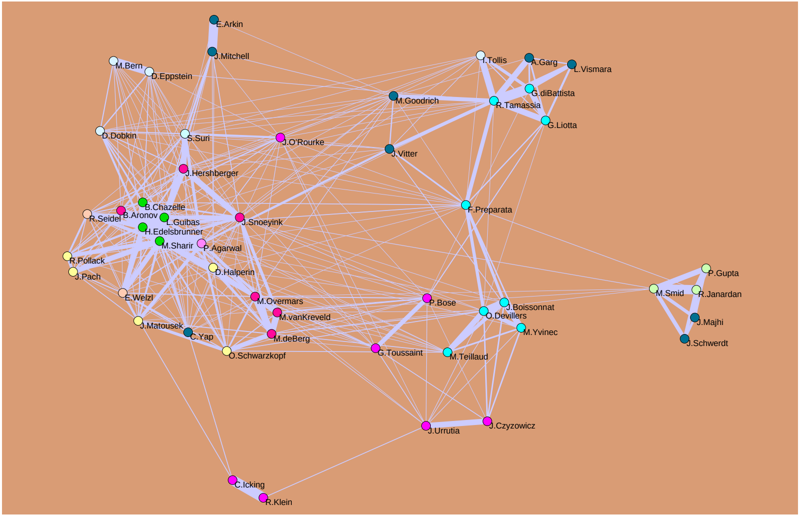
The degrees and the functions p_S, p_M and p_k are monotone. For a monotone function the p-core at level t can be determined, as in the ordinary case, by successively deleting nodes with value of p lower than t; and the cores on different levels are nested

$$t_1 < t_2 \Rightarrow \mathbf{H}_{t_2} \subseteq \mathbf{H}_{t_1}$$

The p-function is *local* iff $p(v, U) = p(v, N_U(v))$.

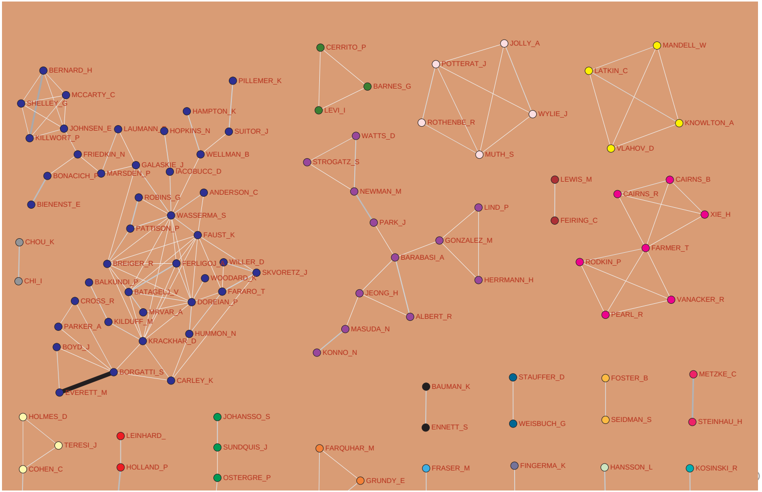
The degrees, p_S and p_M are local; but p_k is **not** local for $k \geq 4$. For a local p-function an $O(m \max(\Delta, \log n))$ algorithm for determining the p-core levels exists, assuming that $p(v, N_C(v))$ can be computed in $O(\deg_C(v))$.

p_S-core at level 46 of Geombib network



p_S-core at level 0.75 in **Cn**(SN5)

Network SN5 (2008): for "social network*" + most frequent references + around 100 social networkers; $|W| = 193376$, $|C| = 7950$, $|A| = 75930$, $|J| = 14651$, $|K| = 29267$



Clustering and blockmodeling

For clustering the nodes, we use the normalized matrices. When computing the dissimilarity $d[u,v]$ between nodes u and v in a weighted network **C** it is important to use the corrected dissimilarities. Here we compare the entry $c[u,t]$ with the entry $c[v,t]$ for t different from u or v, and the entry $c[u,v]$ with the entry $c[v,u]$, and the entry $c[u,u]$ with the entry $c[v,v]$. We selected the *corrected Euclidean distance* (Doreian et al., 2005, p. 181)

$$d_e[u,v] = \sqrt{(c[u,v]-c[v,u])^2 + (c[u,u]-c[v,v])^2 + \sum_{t:t\neq u \wedge t\neq v} (c[u,t]-c[v,t])^2}$$

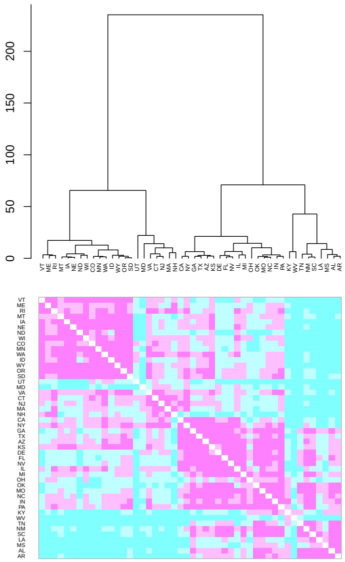
The US geographical data, 2016

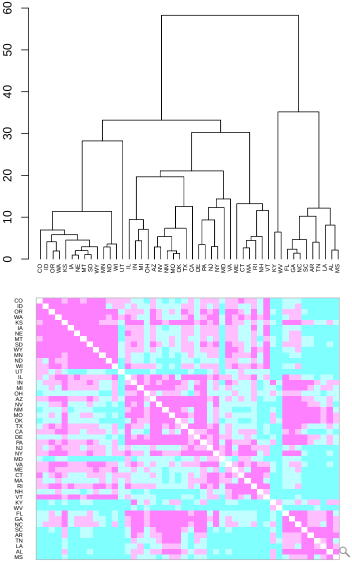
From the site https://datausa.io/profile/geo/united-states/ we obtained the data about US states in 2016 for the following variables: `crime` – homicide deaths, `violent` – violent crimes, `smoking` – adult smoking prevalence, `drinking` – excessive drinking prevalence, `diabetes` – diabetes prevalence, `opioid` – opioid overdose death rate, and `income` – median household income.

In his book *The Stanford GraphBase* D.E. Knuth provided a description of neighboring relation for the contiguous part of USA `contiguous-usa.dat` (without Alaska and Hawai). Because of missing data we removed also Washington DC.

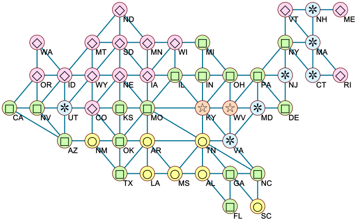
wiki

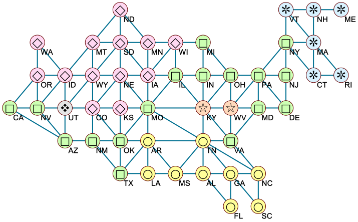
USA: Ward clustering (left) and Maximum/Tolerant clustering (right)





USA: Ward clustering (left) and Maximum/Tolerant clustering (right)





$$
\begin{aligned}
C_1 &= \{AL, AR, LA, MS, NM, TN, SC\},\\
C_2 &= \{AZ, CA, DE, FL, GA, IL, IN, KS, MI, MO,\\
&\qquad NC, NV, NY, OH, OK, PA, TX\},\\
C_3 &= \{CO, IA, ID, ME, MN, MT, ND, NE, OR,\\
&\qquad SD, WY, RI, WI, WA, VT\},\\
C_4 &= \{CT, MA, MD, NH, NJ, UT, VA\},\\
C_5 &= \{KY, WV\}.
\end{aligned}
$$

$$
\begin{aligned}
C_1 &= \{AL, AR, FL, GA, LA, MS, NC, TN, SC\},\\
C_2 &= \{AZ, CA, DE, IL, IN, MD, MI, MO, NJ,\\
&\qquad NM, NV, NY, OH, OK, PA, VA, TX\},\\
C_3 &= \{CO, IA, ID, KS, MN, MT, ND, NE, OR,\\
&\qquad SD, WY, WI, WA\},\\
C_4 &= \{CT, MA, ME, NH, RI, VT\},\\
C_5 &= \{KY, WV\},\\
C_6 &= \{UT\}.
\end{aligned}
$$

Averages for Maximum/Tolerant clustering

	crime	*violent*	*smoking*	*drinking*	*diabetes*	*opioid*	*income*
C_1	8.1667	462.00	0.2140	0.1488	0.1160	10.788	46104
C_2	5.9701	425.91	0.1804	0.1719	0.1032	15.794	57054
C_3	2.8385	265.25	0.1765	0.2005	0.0852	7.408	55913
C_4	2.3833	234.31	0.1660	0.1932	0.0880	26.717	62751
C_5	4.9000	273.02	0.2645	0.1195	0.1210	33.500	43727
C_6	1.9000	204.72	0.0970	0.1210	0.0710	16.400	62518
all	4.9563	354.23	0.1856	0.1748	0.0989	14.700	54963
C_1	**1.3181**	**0.8278**	0.8162	−0.8584	1.1929	−0.4304	−1.0281
C_2	0.4165	0.5506	−0.1502	−0.0928	0.3026	0.1204	0.2427
C_3	−0.8695	−0.6836	−0.2620	**0.8523**	−0.9584	−**0.8024**	0.1103
C_4	−1.0564	−0.9212	−0.5625	0.6087	−0.7599	1.3223	**0.9039**
C_5	−0.0231	−0.6239	**2.2668**	−**1.8260**	**1.5416**	**2.0687**	−**1.3039**
C_6	−**1.2548**	−**1.1485**	−**2.5445**	−1.7764	−**1.9456**	0.1871	0.8767

References I

- De Nooy, W, Mrvar, A, Batagelj, V: Exploratory Social Network Analysis with Pajek; Revised and Expanded Edition for Updated Software. Structural Analysis in the Social Sciences, CUP, July 2018.
- Batagelj, V, Doreian, P, Ferligoj, A, Kejžar, N: Understanding Large Temporal Networks and Spatial Networks: Exploration, Pattern Searching, Visualization and Network Evolution. Wiley Series in Computational and Quantitative Social Science. Wiley, October 2014
- Batagelj, V, Zaveršnik, M: Fast algorithms for determining (generalized) core groups in social networks. Advances in Data Analysis and Classification, 2011. Volume 5, Number 2, 129-145.
- Batagelj, V, Cerinšek, M: On bibliographic networks. Scientometrics 96 (2013) 3, 845-864.

References II

- Cerinšek, M, Batagelj, V: Sources of Network Data. Encyclopedia of Social Network Analysis and Mining. Reda Alhajj, Jon Rokne (Eds.), Springer, 2017
- Batagelj, V: On fractional approach to analysis of linked networks. Scientometrics 123 (2020) 2: 621-633
- Batagelj, V, Maltseva, D: Temporal bibliographic networks. Journal of Informetrics, Volume 14, Issue 1, February 2020, 101006.
- Maltseva, D, Batagelj, V: Social network analysis as a field of invasions: bibliographic approach to study SNA development. Scientometrics, 121(2019)2, 1085-1128
- Schvaneveldt, RW (Ed.) (1990) Pathfinder Associative Networks: Studies in Knowledge Organization. Norwood, NJ: Ablex.
- Wikipedia: Regular language.