



Info

V. Batagelj

Teacher

Program

Requirements

Time-table

Sources

Examples

Introduction to Network Analysis using **Pajek**

0. Info

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

PhD and MS program in Statistics
University of Ljubljana, 2022



Outline

Info

V. Batagelj

Teacher

Program

Requirements

Time-table

Sources

Examples

- 1 Teacher
- 2 Program
- 3 Requirements
- 4 Time-table
- 5 Sources
- 6 Examples



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

Current version of slides (February 17, 2022 at 01 :19): [slides PDF](#)



Teacher

Info

V. Batagelj

Teacher

Program

Requirements

Time-table

Sources

Examples

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

Wiki page:

<http://vladowiki.fmf.uni-lj.si/doku.php?id=pajek:ev:pd:p22>

Course official page:

<https://e.fe.uni-lj.si/course/view.php?id=575>



Requirements/Exam

Info

V. Batagelj

Teacher

Program

Requirements

Time-table

Sources

Examples

Home project:

- ① for a network given in a picture make its description on a file and make a visualization an basic analyses.
- ② transform given data into network and analyze it
- ③ analysis of a large network



Resources

Info

V. Batagelj

Teacher

Program

Requirements

Time-table

Sources

Examples

- Pajek: <http://mrvar.fdv.uni-lj.si/pajek/>
- igraph R package: <http://igraph.org/r/>
- Inkscape: <https://inkscape.org/en/>
- Ghostscript, Ghostview and GSview:
<http://pages.cs.wisc.edu/~ghost/>



Roman road system

Tabula Peutingeriana

Info

V. Batagelj

Teacher

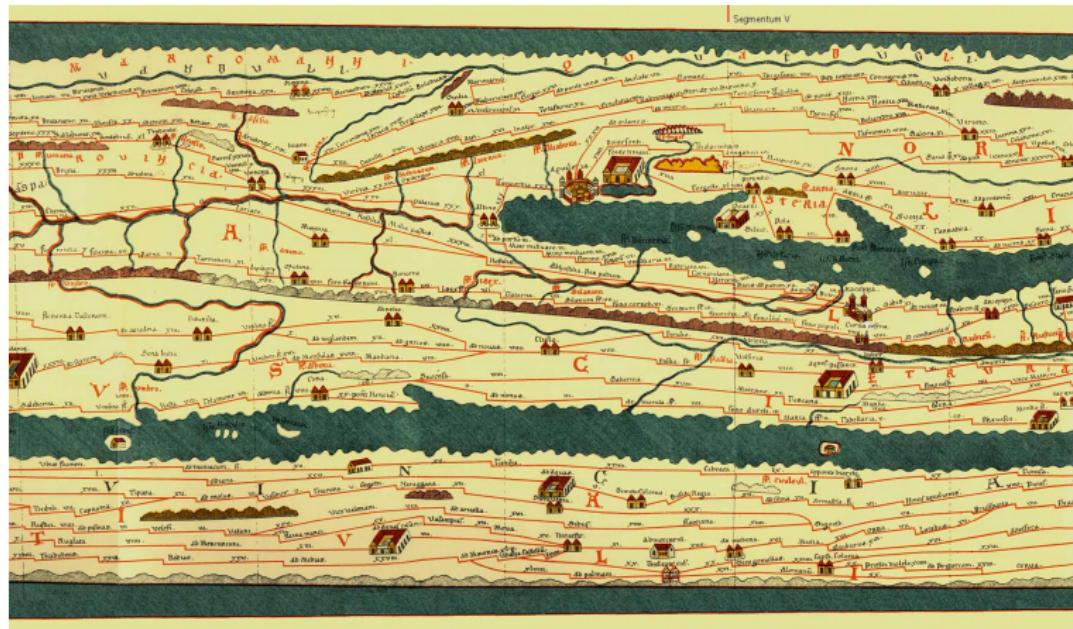
Program

Requirements

Time-table

Sources

Examples





Flowcharts

McCabe

Info

V. Batagelj

Teacher

Program

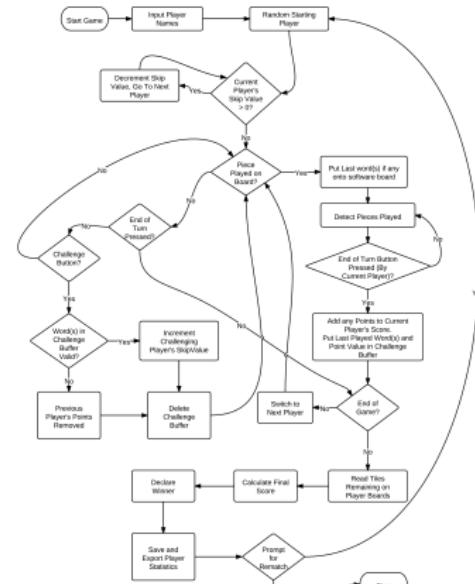
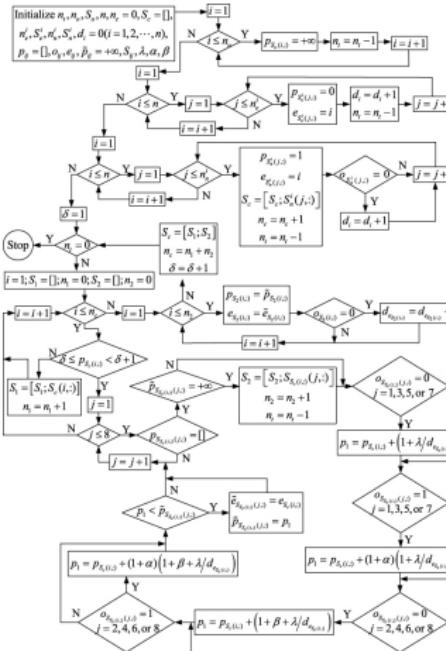
Requirements

Time-table

Sources

Examples

IOP





Large organic molecules

3CRO, PDB

Info

V. Batagelj

Teacher

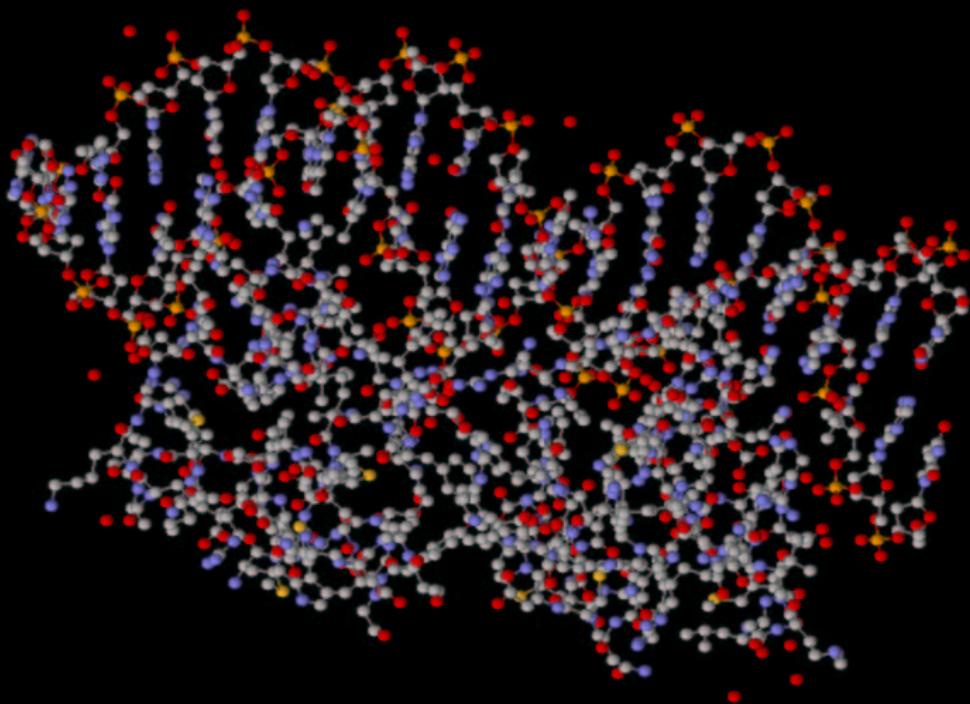
Program

Requirements

Time-table

Sources

Examples





Protein-protein interaction network

PluriPlus

Info

V. Batagelj

Teacher

Program

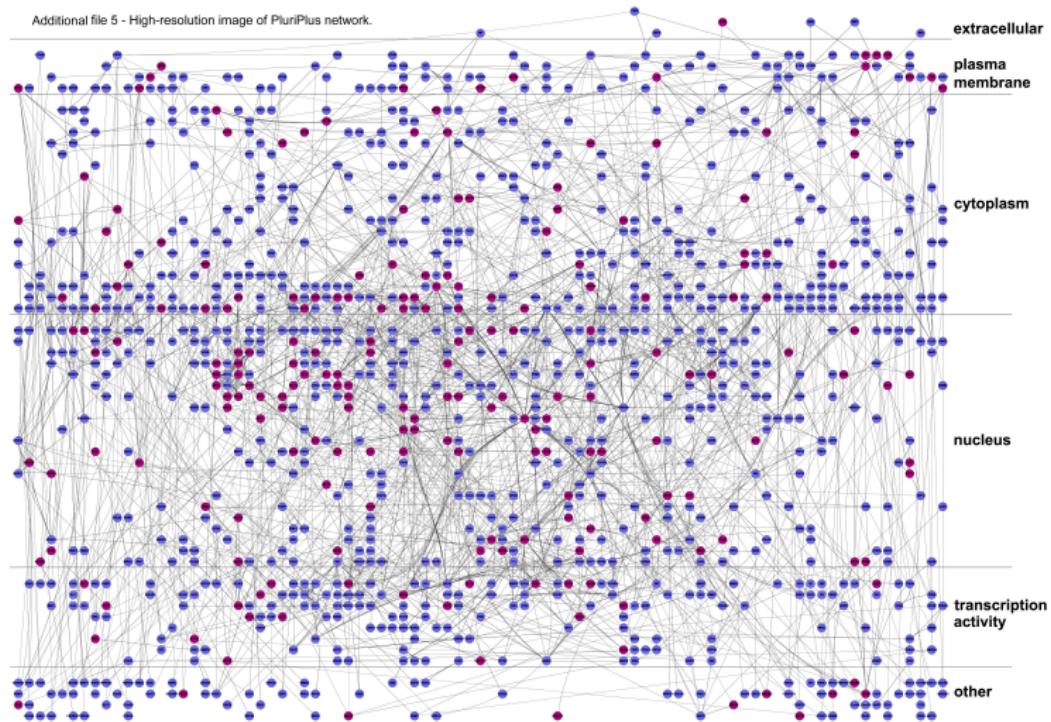
Requirements

Time-table

Sources

Examples

Additional file 5 - High-resolution image of PluriPlus network.





Austrian research projects collaboration

FAS

Info

V. Batagelj

Teacher

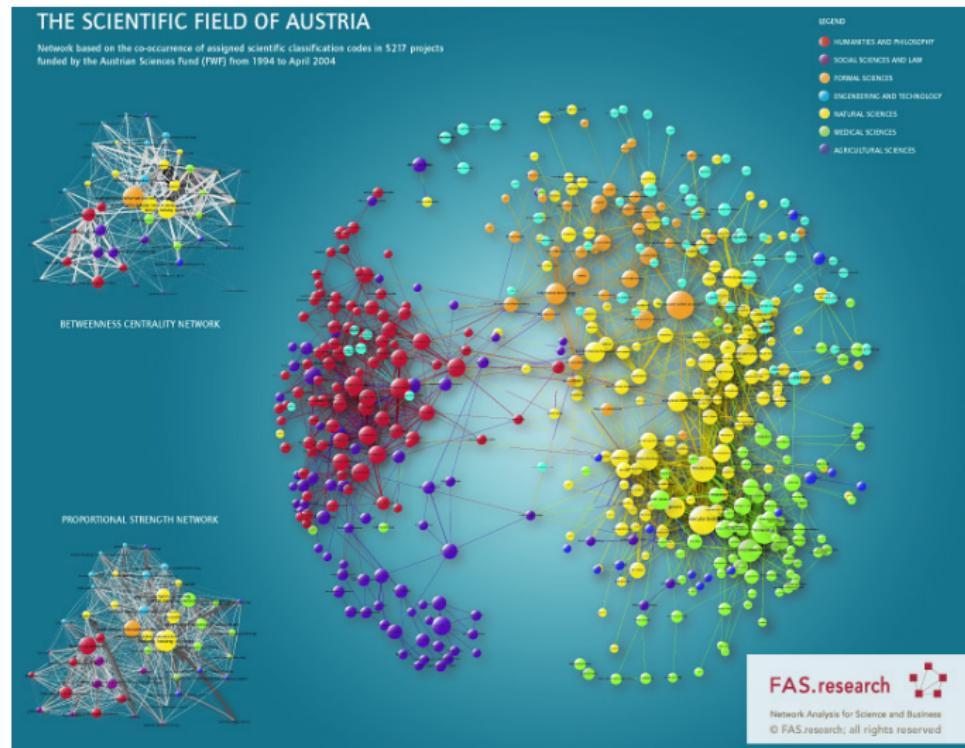
Program

Requirements

Time-table

Sources

Examples





Internet

Cheswick, Opte

Info

V. Batagelj

Teacher

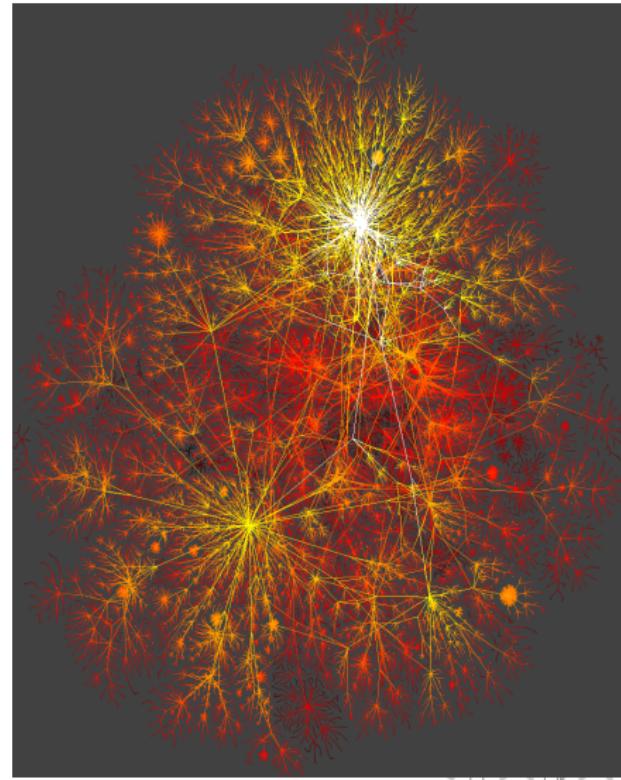
Program

Requirements

Time-table

Sources

Examples





They Rule

Info

V. Batagelj

Teacher

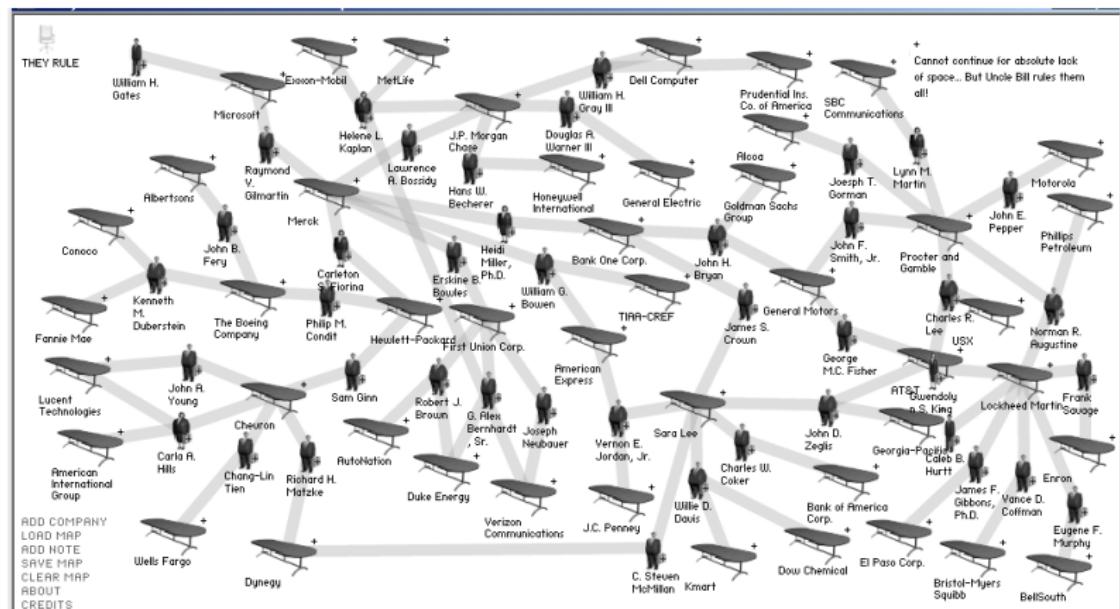
Program

Requirements

Time-table

Sources

Examples





"Countryside" school district

Info

V. Batagelj

Teacher

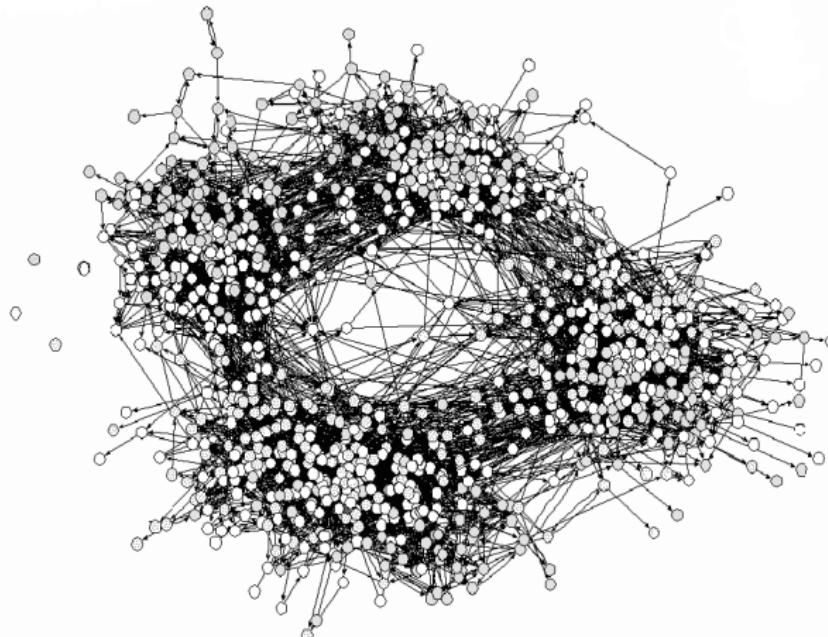
Program

Requirements

Time-table

Sources

Examples



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



Mark Lombardi

Google

Info

V. Batagelj

Teacher

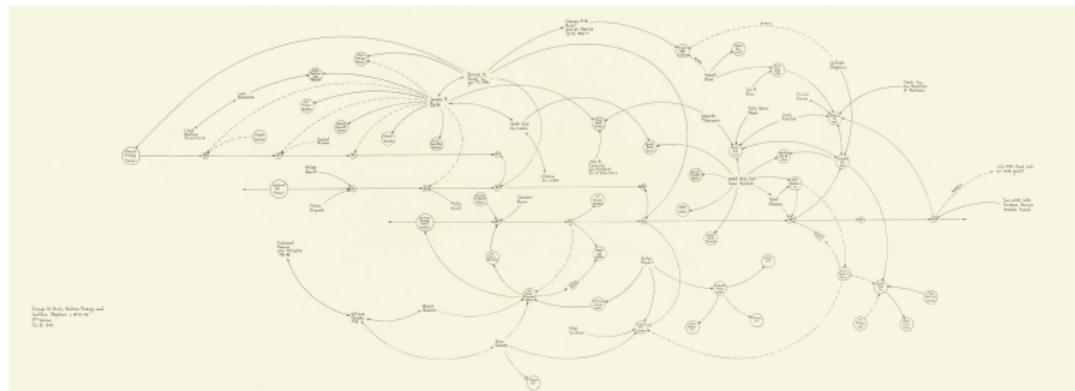
Program

Requirements

Time-table

Sources

Examples



More examples at [Visual Complexity](#)



Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

Introduction to Network Analysis using **Pajek**

1. Description of networks

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

PhD and MS program in Statistics
University of Ljubljana, 2022



Outline

Description

V. Batagelj

Networks

Descriptions of networks

Properties

Types of networks

Temporal networks

Two-mode networks

Multi-relational networks

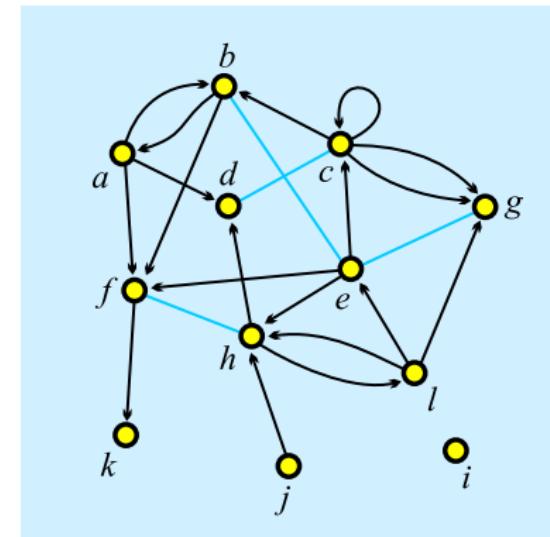
Two-mode networks

igraph in R

Pajek and R

netsJSON and Nets

- 1 Networks
- 2 Descriptions of networks
- 3 Properties
- 4 Types of networks
- 5 Temporal networks
- 6 Multi-relational networks
- 7 Two-mode networks
- 8 igraph in R
- 9 Pajek and R
- 10 netsJSON and Nets



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

Current version of slides (February 17, 2022 at 01 :55): [slides PDF](#)



Networks

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

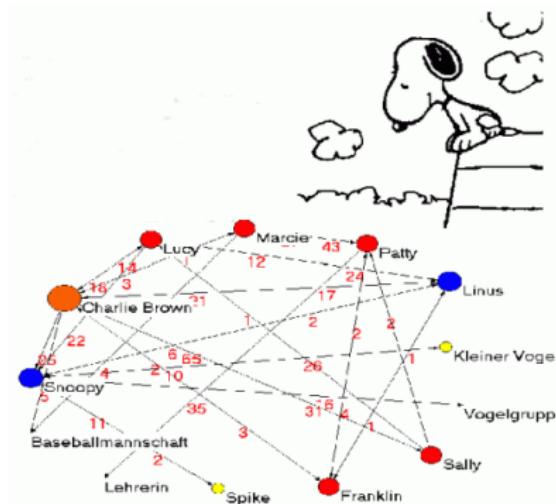
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



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

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

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 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: $p: \mathcal{V} \rightarrow A$
- \mathcal{W} **link value functions** / weights: $w: \mathcal{L} \rightarrow B$



Graph

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

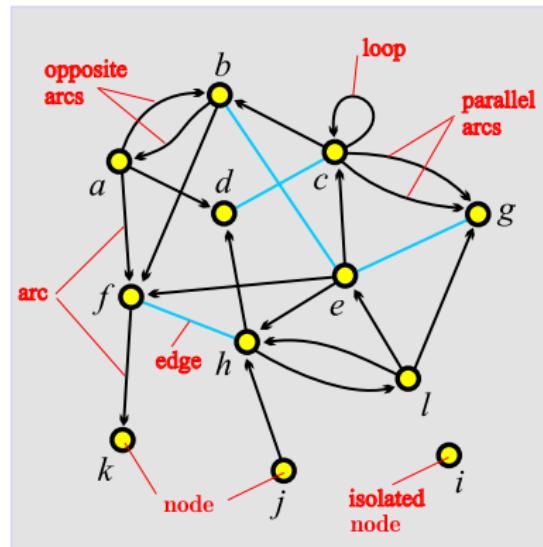
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



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.



ESNA Pajek

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

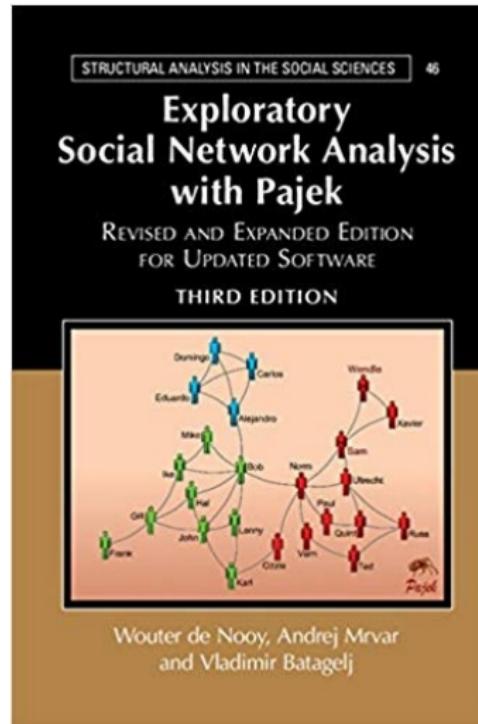
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



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/>



igraph

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

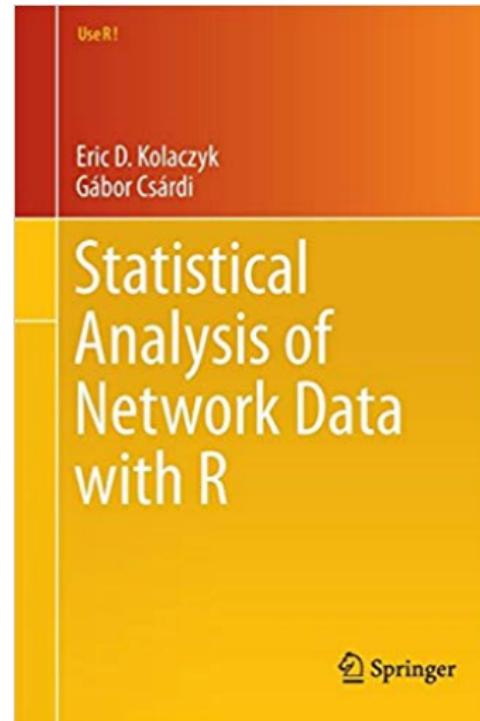
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



A book on Statistical Analysis of Network Data with R using the package igraph was written by Kolaczyk, Eric D. and Csárdi, Gábor (Springer 2014).

Another book on igraph is prepared by Gábor Csárdi, Tamás Nepusz and Edoardo M. Airoldi draft.

igraph can be installed from CRAN

<https://cran.r-project.org/web/packages/igraph/index.html>



Graph / Sets – NET

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

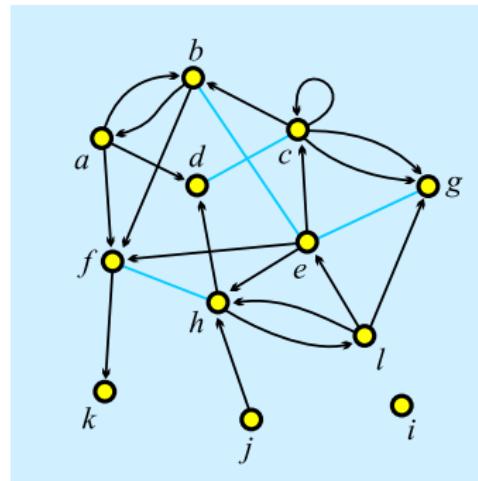
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



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

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

Pajek: local: `GraphSet`; `TinaSet`;

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



Graph / Sets – NET

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

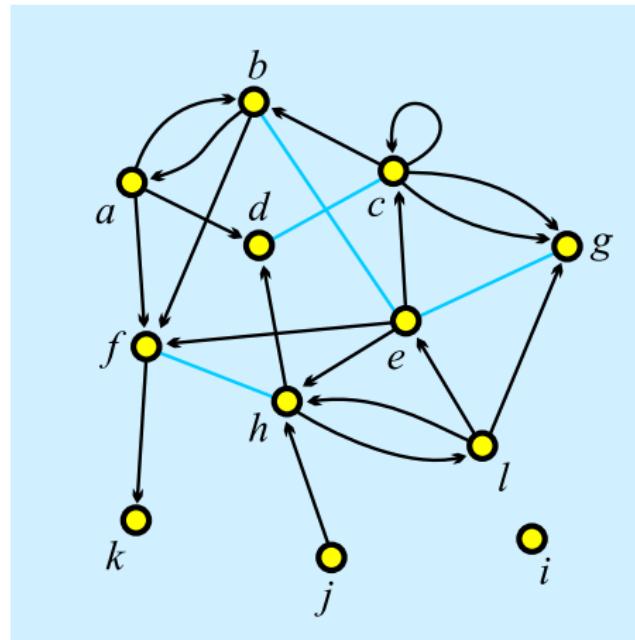
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



*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 / Neighbors – NET

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

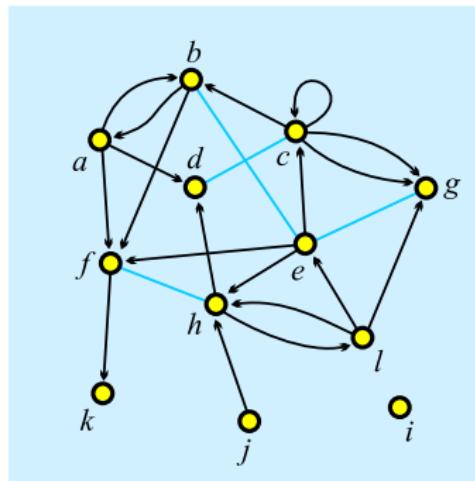
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



$$N_A(a) = \{b, d, f\}$$

$$N_A(b) = \{a, f\}$$

$$N_A(c) = \{b, c, g, h\}$$

$$N_A(e) = \{c, f, h\}$$

$$N_A(f) = \{k\}$$

$$N_A(h) = \{d, l\}$$

$$N_A(j) = \{h\}$$

$$N_A(l) = \{e, g, h\}$$

$$N_E(e) = \{b, g\}$$

$$N_E(c) = \{d\}$$

$$N_E(f) = \{h\}$$

Pajek: local: `GraphList`; `TinaList`;

WWW: `GraphList / net`; `TinaList / net`.

$$N(v) = N_A(v) \cup N_E(v), \quad \text{also} \quad N_{out}(v), \ N_{in}(v)$$

Star in v , $S(v)$ is the set of all links with v as their initial node.



Graph / Neighbors – NET

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

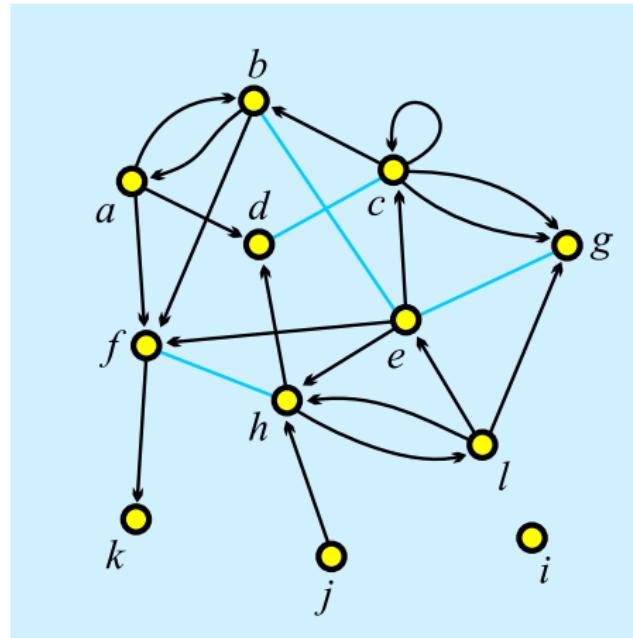
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



*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

*Arcslist

1	2	4	6
2	1	6	
3	2	3	7
5	3	6	8
6	11		
8	4	12	
10	8		
12	5	7	8

*Edgeslist

2	5
3	4
5	7
6	8



Graph / Matrix – MAT

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

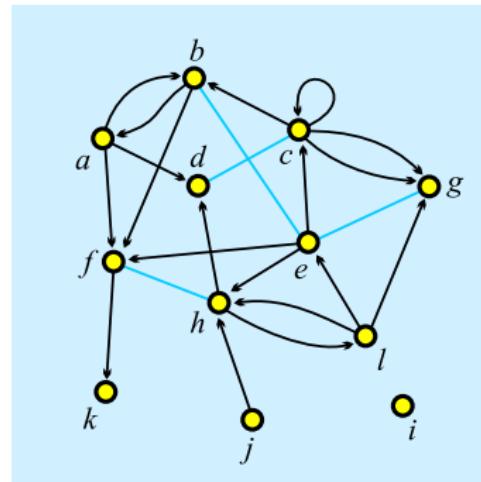
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



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

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

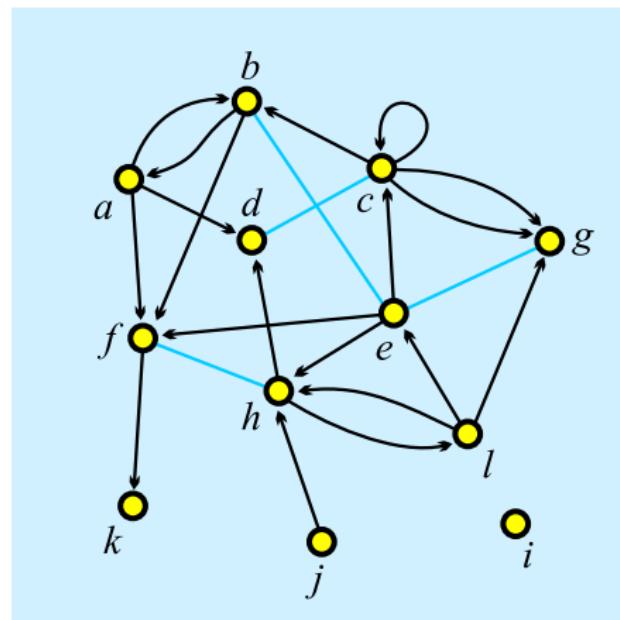
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



*Vertices	12
1 "a"	0.1020
2 "b"	0.2860
3 "c"	0.5322
4 "d"	0.3259
5 "e"	0.5543
6 "f"	0.1552
7 "g"	0.8293
8 "h"	0.4479
9 "i"	0.8204
10 "j"	0.4789
11 "k"	0.1175
12 "l"	0.7095

*Matrix	0 1 0 1 0 1 0 0 0 0 0 0
0 1 0 0 0 1 1 0 0 0 0 0 0	1 0 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 1 0 0 1 1 1 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 1 0 0 1 0
0 1 1 0 0 1 1 1 0 0 0 0 0	0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1 0 0 0 1 0	0 0 0 0 1 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 1 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 1 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 1 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 1 0 0 0
0 0 0 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 1 0 1 1 0 0 0 0	0 0 0 0 0 1 0 1 1 0 0 0



Node Properties / CLU, VEC, PER

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

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

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

inter.net	inter.net	sex.clu	age.vec	rank.per
*Vertices 20		*vertices 20	*vertices 20	*vertices 20
1 "m01"	1 6 5	1	15	1
2 "m02"	1 7 9	1	10	2
3 "m03"	1 8 7	1	10	3
4 "m04"	1 9 4	1	8	4
5 "m05"	1 10 3	1	7	5
6 "f06"	1 11 3	2	15	10
7 "f07"	1 12 7	2	5	11
8 "f08"	1 13 3	2	11	6
9 "f09"	1 14 2	2	8	12
10 "f10"	1 15 5	2	9	9
11 "f11"	1 16 1	2	16	7
12 "f12"	1 17 4	2	10	8
13 "f13"	1 18 1	2	14	18
14 "f14"	2 3 5	2	5	19
15 "f15"	2 4 1	2	7	20
16 "f16"	2 5 3	2	11	13
17 "f17"	2 6 1	2	7	14
18 "f18"	2 7 4	2	5	15
19 "f19"	2 8 2	2	15	16
20 "f20"	2 9 6	2	4	17
*Edges	2 10 2			
1 2 2	2 11 5			
1 3 10	2 12 4			
1 4 4	2 13 3			
- - -	2 14 2			
	...			

Important note: 0 is not allowed as node number.



Pajek's Project File / PAJ

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

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 ([PDF/paj](#)).



Special graphs – path, cycle, star, complete

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

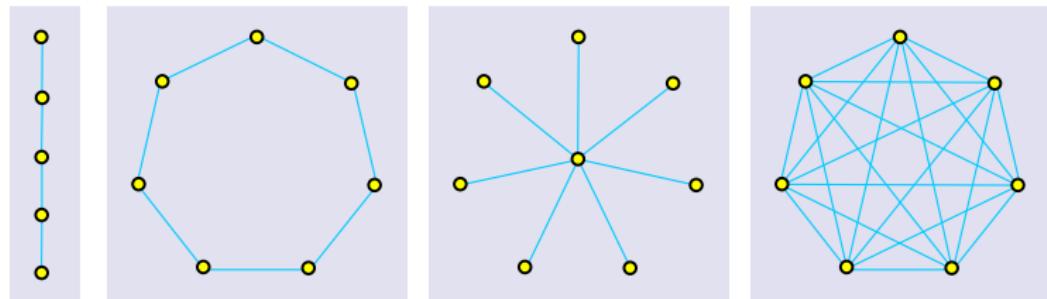
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



Graphs: *path P_5 , cycle C_7 , star S_8 in complete graph K_7 .*



Representations of properties

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

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 *vectors*, nominal properties by *partitions* or as *labels* 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 values can be assigned as *label*, *color* or *line pattern* (see **Pajek manual**, section **4.3**).



Some related operations

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

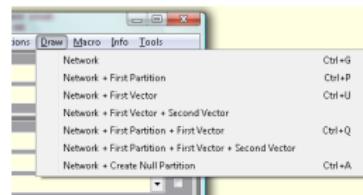
Multi-
relational
networks

Two-mode
networks

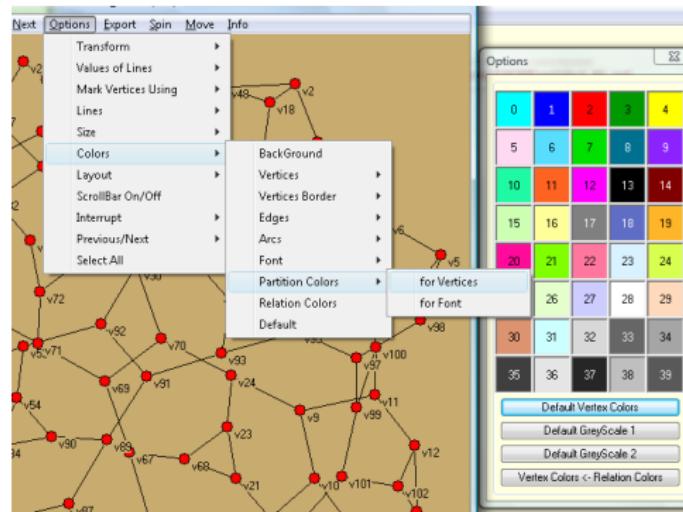
igraph in R

Pajek and R

netsJSON
and Nets



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





Display of properties – school (Moody)

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

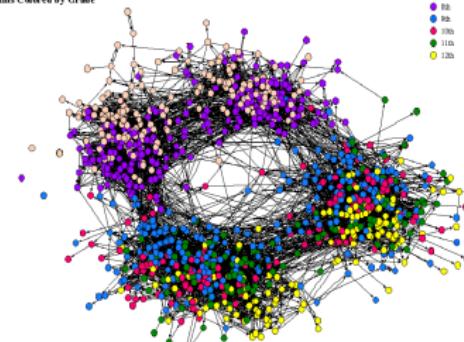
igraph in R

Pajek and R

netsJSON
and Nets

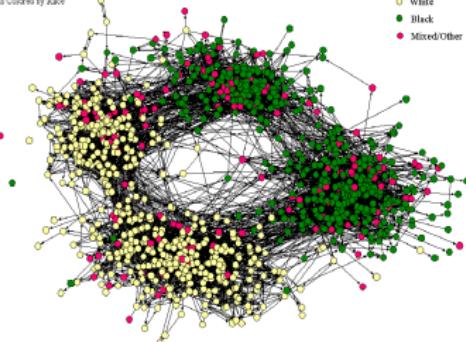
The Social Structure of "Countryside" School District

Points Colored by Grade



The Social Structure of "Countryside" School District

Points Colored by Race





Types of networks

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

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.
- *multi-relational networks*.
- *temporal networks*, dynamic graphs – networks changing over time.
- 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).



Temporal networks

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

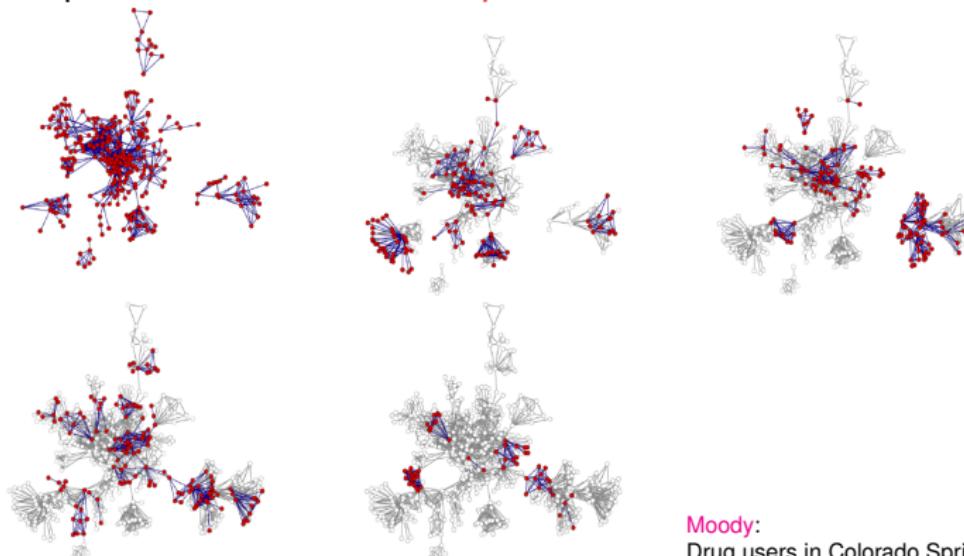
Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

In a *temporal network* the presence/activity of node/link can change through time. **Pajek** supports two types of descriptions of temporal networks based on *presence* and on *events*.



Moody:
Drug users in Colorado Springs, 5
years



Temporal network

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

Temporal network

$$\mathcal{N}_T = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W}, T)$$

is obtained if the *time* T is attached to an ordinary network. T is a set of *time points* $t \in T$.

In temporal network nodes $v \in \mathcal{V}$ and links $l \in \mathcal{L}$ are not necessarily present or active in all time points. If a link $l(u, v)$ is active in time point t then also its endnodes u and v should be active in time t .

We will denote the network consisting of links and nodes active in time $t \in T$ by $\mathcal{N}(t)$ and call it a *time slice* in time point t . To get time slices in **Pajek** use

Network/Temporal Network/Generate in time



Temporal networks – presence

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

```
*Vertices 3
1 "a" [5-10,12-14]
2 "b" [1-3,7]
3 "e" [4-*]
*Edges
1 2 1 [7]
1 3 1 [6-8]
```

Node a is present in time points 5, 6, 7, 8, 9, 10 and 12, 13, 14.

Edge (1 : 3) is present in time points 6, 7, 8.

* means 'infinity'.

A link is present, if both its endnodes are present.

Time.net
netsJSON



Temporal networks – events

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

Event	Explanation
TI t	initial events – following events happen when time point t starts
TE t	end events – following events happen when time point t is finished
AV $v \ n \ s$	add vertex v with label n and properties s
HV v	hide node v
SV v	show node v
DV v	delete node v
AA $u \ v \ s$	add arc (u, v) with properties s
HA $u \ v$	hide arc (u, v)
SA $u \ v$	show arc (u, v)
DA $u \ v$	delete arc (u, v)
AE $u \ v \ s$	add edge $(u : v)$ with properties s
HE $u \ v$	hide edge $(u : v)$
SE $u \ v$	show edge $(u : v)$
DE $u \ v$	delete edge $(u : v)$
CV $v \ s$	change property of node v to s
CA $u \ v \ s$	change property of arc (u, v) to s
CE $u \ v \ s$	change property of edge $(u : v)$ to s
CT $u \ v$	change (un)directedness of link (u, v)
CD $u \ v$	change direction of arc (u, v)
PE $u \ v \ s$	replace pair of arcs (u, v) and (v, u) by single edge $(u : v)$ with properties s
AP $u \ v \ s$	add pair of arcs (u, v) and (v, u) with properties s
DP $u \ v$	delete pair of arcs (u, v) and (v, u)
EP $u \ v \ s$	replace edge $(u : v)$ by pair of arcs (u, v) and (v, u) with properties s

s can be empty.

In case of parallel links : k denotes the k -th link – HE:3 14 37 hides the third edge linking nodes 14 and 37.

*Vertices 3

*Events

TI	1
AV	2 "b"
TE	3
HV	2
TI	4
AV	3 "e"
TI	5
AV	1 "a"
TI	6
AE	1 3 1
TI	7
SV	2
AE	1 2 1
TE	7
DE	1 2
DV	2
TE	8
DE	1 3
TE	10
HV	1
TI	12
SV	1
TE	14
DV	1

Time.tim Friends.tim.

File/Network/Read Time Events



Temporal networks / September 11

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

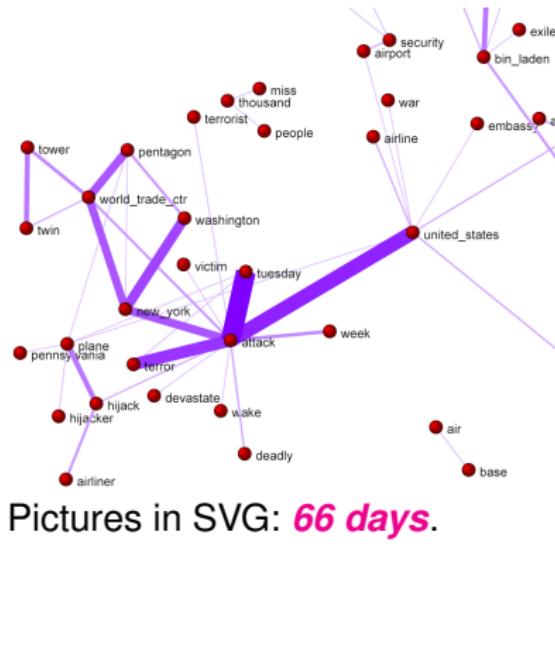
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



Pictures in SVG: **66 days**.

Steve Corman with collaborators from Arizona State University transformed, using his Centering Resonance Analysis (**CRA**), daily Reuters news (66 days) about September 11th into a temporal network of words coappearance.



Multi-relational networks

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

A *multi-relational network* is denoted by

$$\mathcal{N} = (\mathcal{V}, (\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_k), \mathcal{P}, \mathcal{W})$$

and contains different relations \mathcal{L}_i (sets of links) over the same set of nodes. Also the weights from \mathcal{W} are defined on different relations or their union.

Examples of such networks are: Transportation system in a city (stations, lines); **WordNet** (words, semantic relations: synonymy, antonymy, hyponymy, meronymy, ...), **KEDS** networks (states, relations between states: Visit, Ask information, Warn, Expel person, ...), ...



... Multi-relational networks

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R
netsJSON
and Nets

The relation can be assigned to a link as follows:

- add to a keyword for description of links (`*arcs`, `*edges`, `*arcslist`, `*edgeslist`, `*matrix`) the number of relation followed by its name:

```
*arcslist :3 "sent a letter to"
```

All links controlled by this keyword belong to the specified relation. (`Sampson`, `SampsonL`)

- Any link controlled by `*arcs` or `*edges` can be assigned to selected relation by starting its description by the number of this relation.

```
3: 47 14 5
```

Link with endnodes 47 and 14 and weight 5 belongs to relation 3.



Computer-assisted text analysis

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

An often used way to obtain networks is the *computer-assisted text analysis* (CaTA).

Terms considered in TA are collected in a *dictionary* (it can be fixed in advance, or built dynamically). The main two problems with terms are *equivalence* (different words representing the same term) and *ambiguity* (same word representing different terms). Because of these the *coding* – transformation of raw text data into formal *description* – is done often manually or semiautomatically. As *units* of TA we usually consider clauses, statements, paragraphs, news, messages, ...

Till now the thematic and semantic TA mainly used statistical methods for analysis of the coded data.

In thematic TA the units are coded as rectangular matrix *Text units* \times *Concepts* which can be considered as a two-mode network.

Examples: M.M. Miller: VBPro, H. Klein: Text Analysis/ TextQuest.



... approaches to CaTA

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

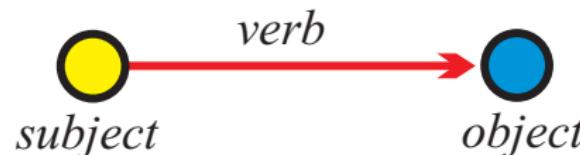
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



Examples: Roberto Franzosi; KEDS, Tabari, KEDS / Gulf.

This coding can be directly considered as network with *Subjects* \cup *Objects* as nodes and links (arcs) labeled with *Verbs*.

See also RDF triples in semantic web, SPARQL.



Multi-relational temporal network – KEDS/WEIS

Description

V. Batagelj

Networks

Descriptions of networks

Properties

Types of networks

Temporal networks

Multi-relational networks

Two-mode networks

igraph in R

Pajek and R

netsJSON and Nets

```
% 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 ET
224: 3 83 1 [4] 890407 ALB ETHALB 224 (RIOT) RIOTS
123: 83 153 1 [4] 890408 ETHALB KSV 123 (INVESTIGATE) PROBIN

...
42: 105 63 1 [175] 030731 GER CYP 042 (ENDORSE) GAVE S
212: 295 35 1 [175] 030731 UNWCT BOSSER 212 (ARREST PERSON) SENTEN
43: 306 87 1 [175] 030731 VAT EUR 043 (RALLY) RALLIED
13: 295 35 1 [175] 030731 UNWCT BOSSER 013 (RETRACT) CLEARE
121: 295 22 1 [175] 030731 UNWCT BAL 121 (CRITICIZE) CHARGE
122: 246 295 1 [175] 030731 SER UNWCT 122 (DENIGRATE) TESTIF
121: 35 295 1 [175] 030731 BOSSER UNWCT 121 (CRITICIZE) ACCUSE
```

Kansas Event Data System *KEDS*



Two-mode networks

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

In a *two-mode* network $\mathcal{N} = ((\mathcal{U}, \mathcal{V}), \mathcal{L}, \mathcal{P}, \mathcal{W})$ the set of nodes consists of two disjoint sets of nodes \mathcal{U} and \mathcal{V} , and all the links from \mathcal{L} have one endnode in \mathcal{U} and the other in \mathcal{V} . Often also a *weight* $w : \mathcal{L} \rightarrow \mathbb{R} \in \mathcal{W}$ is given; if not, we assume $w(u, v) = 1$ for all $(u, v) \in \mathcal{L}$.

A two-mode network can also be described by a rectangular matrix $\mathbf{A} = [a_{uv}]_{\mathcal{U} \times \mathcal{V}}$.

$$a_{uv} = \begin{cases} w_{uv} & (u, v) \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases}$$

Examples: (persons, societies, years of membership), (buyers/consumers, goods, quantity), (parliamentarians, problems, positive vote), (persons, journals, reading).

A two-mode network is announced by \star vertices $n n_{\mathcal{U}}$.

Authors and works.



Deep South

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



Classical example of two-mode network are the Southern women (Davis 1941).

[Davis.paj](#). Freeman's overview.

NAMES OF PARTICIPANTS OF GROUP I	CODE NUMBERS AND DATES OF SOCIAL EVENTS REPORTED IN <i>Old City Herald</i>													
	(1) 6/27	(2) 3/2	(3) 4/12	(4) 9/26	(5) 2/25	(6) 5/19	(7) 3/15	(8) 9/16	(9) 4/6	(10) 8/10	(11) 2/23	(12) 6/7	(13) 11/21	(14) 8/3
1. Mrs. Evelyn Jefferson.....	X	X	X	X	X	X	X	X
2. Miss Laura Mandeville.....	X	X	X	X	X	X	X	X
3. Miss Theresa Anderson.....	X	X	X	X	X	X	X	X	X	X	X
4. Miss Brenda Rogers.....	X	X	X	X	X	X	X	X
5. Miss Charlotte McDowell.....	X	X	X	X
6. Miss Frances Anderson.....	X	X	X	X	X
7. Miss Eleanor Nye.....	X	X	X	X	X	X	X	X
8. Miss Pearl Oglethorpe.....	X	X	X	X	X	X	X
9. Miss Ruth DeSand.....	X	X	X	X	X	X
10. Miss Verne Sanderson.....	X	X	X	X	X	X
11. Miss Myra Liddell.....	X	X	X	X	X	X	X
12. Miss Katherine Rogers.....	X	X	X	X	X	X	X
13. Mrs. Sylvia Avondale.....	X	X	X	X	X	X
14. Mrs. Nora Fayette.....	X	X	X	X	X	X	X
15. Mrs. Helen Lloyd.....	X	X	X	X	X	X	X
16. Mrs. Dorothy Murchison.....	X	X	X	X	X	X	X
17. Mrs. Olivia Carlton.....	X	X	X	X	X	X
18. Mrs. Flora Price.....	X	X	X	X	X



igraph Example

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

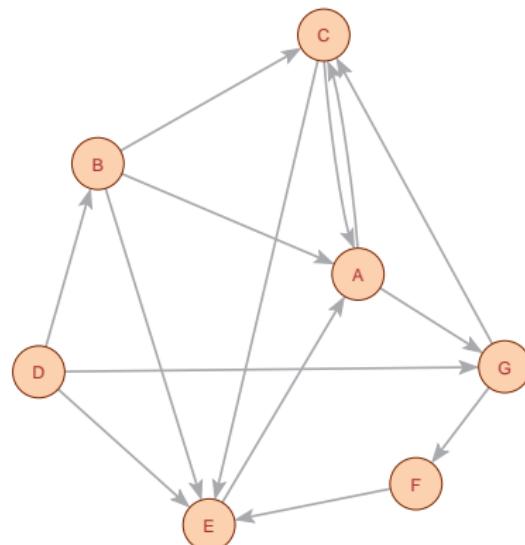
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets



```
> library(igraph)
> links <- c("A", "C", "A", "G",
+ "B", "C", "B", "A", "B", "E",
+ "C", "A", "C", "E", "D", "B",
+ "D", "G", "D", "E", "E", "A",
+ "F", "E", "G", "C", "G", "F")
> L <- graph(links)
> L
IGRAPH bb7e45b DN-- 7 14 --
+ attr: name (v/c)
+ edges from bb7e45b (vertex names)
[1] A->C A->G B->C B->A B->E C->A
> plot(L)
> vcount(L)
[1] 7
> ecount(L)
[1] 14
> L <- L + vertex("H")
> plot(L)
```

igraph is a library for analyzing networks. It has also an R interface.
For other R libraries for solving network analysis problems see: Ian McCulloch,
Alexander Perrone: R Packages for Social Network Analysis. [ESNAM](#). Springer
2018.

See also: [sna](#), [network](#), [statnet](#), [ggnetwork](#)



igraph object display

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

D/U ‘D’ – directed / ‘U’ – undirected.

N/- ‘N’ – named (labeled). A dash means that the network is not named.

W/- ‘W’ – weighted (has values on links). Unweighted networks have a dash in this position.

B/- ‘B’ – bipartite (two-mode). A dash means that the network is one-mode.



igraph attributes

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

```
> V(L)
+ 8/8 vertices, named, from 84e744b:
[1] A C G B E D F H
> E(L)
+ 14/14 edges from 84e744b (vertex names):
[1] A->C A->G B->C B->A B->E C->A C->E D->B D->G D->E E->A F->E
> V(L)$name
[1] "A" "C" "G" "B" "E" "D" "F" "H"
> V(L)$name[5] <- "John"
> V(L)$color <- sample(c("yellow", "cyan"), vcount(L), rep=TRUE)
> plot(L)
> ye <- V(L)[color=="yellow"]; cy <- V(L)[color=="cyan"]
> E(L)[ye %--% cy]$color <- "red"
> E(L)[ye %--% ye]$color <- "blue"
> E(L)[cy %--% cy]$color <- "blue"
> L$name <- "Example"
> E(L)$weight <- sample(1:10, ecount(L), rep=TRUE)
> graph_attr_names(L)
[1] "name"
> graph_attr(L)
$name
[1] "Example"
> vertex_attr_names(L)
[1] "name" "color"
> edge_attr_names(L)
[1] "color" "weight"
> w <- E(L)$weight; plot(L, edge.width=w)
> write.graph(L, "Links.net", format="pajek")
```



Description of networks using a spreadsheet

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

How to describe a network \mathcal{N} ? In principle the answer is simple – we list its components \mathcal{V} , \mathcal{L} , \mathcal{P} , and \mathcal{W} .

The simplest way is to describe a network \mathcal{N} by providing $(\mathcal{V}, \mathcal{P})$ and $(\mathcal{L}, \mathcal{W})$ in a form of two tables.

As an example, let us describe a part of network determined by the following works:

[Generalized blockmodeling](#), [Clustering with relational constraint](#),
[Partitioning signed social networks](#), [The Strength of Weak Ties](#)

There are nodes of different types (modes): persons, papers, books, series, journals, publishers; and different relations among them: author_of, editor_of, contained_in, cites, published_by.

Both tables are often maintained in Excel. They can be exported as text in [CSV](#) (Comma Separated Values) format.



bibNodes.csv

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

name;mode;country;sex;year;vol;num;fPage;lPage;x;y
"Batagelj, Vladimir";person;SI;m;;;;;809.1;653.7
"Doreian, Patrick";person;US;m;;;;;358.5;679.1
"Ferligoj, Anuška";person;SI;f;;;;619.5;680.7
"Granovetter, Mark";person;US;m;;;;145.6;660.5
"Moustaki, Irini";person;UK;f;;;;783.0;228.0
"Mrvar, Andrej";person;SI;m;;;;478.0;630.1
"Clustering with relational constraint";paper;;;1982;47;;413;420
"The Strength of Weak Ties";paper;;;1973;78;6;1360;1380;111.3;320
"Partitioning signed social networks";paper;;;2009;31;1;1;11;408
"Generalized Blockmodeling";book;;;2005;24;;1;385;533.0;445.9
"Psychometrika";journal;;;;;741.8;086.1
"Social Networks";journal;;;;;321.4;236.5
"The American Journal of Sociology";journal;;;;;111.3;168.9
"Structural Analysis in the Social Sciences";series;;;;;310.4
"Cambridge University Press";publisher;UK;;;;;534.3;238.2
"Springer";publisher;US;;;;;884.6;174.0

bibNodes.csv

In large networks, to avoid the empty cells, we split a network to some subnetworks – a collection.



bibLinks.csv

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

```
from;relation;to
"Batagelj, Vladimir";authorOf;"Generalized Blockmodeling"
"Doreian, Patrick";authorOf;"Generalized Blockmodeling"
"Ferligoj, Anuška";authorOf;"Generalized Blockmodeling"
"Batagelj, Vladimir";authorOf;"Clustering with relational constraint"
"Ferligoj, Anuška";authorOf;"Clustering with relational constraint"
"Granovetter, Mark";authorOf;"The Strength of Weak Ties"
"Granovetter, Mark";editorOf;"Structural Analysis in the Social Sciences"
"Doreian, Patrick";authorOf;"Partitioning signed social networks"
"Mrvar, Andrej";authorOf;"Partitioning signed social networks"
"Moustaki, Irini";editorOf;"Psychometrika"
"Doreian, Patrick";editorOf;"Social Networks"
"Generalized Blockmodeling";containedIn;"Structural Analysis in the Social Sciences"
"Clustering with relational constraint";containedIn;"Psychometrika"
"The Strength of Weak Ties";containedIn;"The American Journal of Sociology"
"Partitioning signed social networks";containedIn;"Social Networks"
"Partitioning signed social networks";cites;"Generalized Blockmodeling"
"Generalized Blockmodeling";cites;"Clustering with relational constraint"
"Structural Analysis in the Social Sciences";publishedBy;"Cambridge University Press"
"Psychometrika";publishedBy;"Springer"
```

bibLinks.csv



Factorization and description of large networks

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

To save space and improve the computing efficiency we often replace values of categorical variables with integers. In R this encoding is called a *factorization*.

We enumerate all possible values of a given categorical variable (coding table) and afterwards replace each its value by the corresponding index in the coding table.

This approach is used in most programs dealing with large networks. Unfortunately the coding table is often a kind of meta-data.



CSV2Pajek.R

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

```
# transforming CSV file to Pajek files
# by Vladimir Batagelj, June 2016
# setwd("C:/Users/batagelj/work/Python/graph/SVG/EUSN")
# colC <- c(rep("character",4),rep("numeric",7)); nas=c("", "NA", "NaN")
colC <- c(rep("character",4),rep("numeric",5)); nas=c("", "NA", "NaN")
nodes <- read.csv2("bibNodes.csv",encoding='UTF-8',colClasses=colC,na.strings=nas)
n <- nrow(nodes); M <- factor(nodes$mode); S <- factor(nodes$sex)
mod <- levels(M); sx <- levels(S); S <- as.numeric(S); S[is.na(S)] <- 0
links <- read.csv2("bibLinks.csv",encoding='UTF-8',colClasses="character")
F <- factor(links$from,levels=nodes$name,ordered=TRUE)
T <- factor(links$to,levels=nodes$name,ordered=TRUE)
R <- factor(links$relation); rel <- levels(R)
net <- file("bib.net","w"); cat('*vertices ',n,'\n',file=net)
clu <- file("bibMode.clu","w"); sex <- file("bibSex.clu","w")
cat('%',file=clu); cat('%',file=sex)
for(i in 1:length(mod)) cat(' ',i,mod[i],file=clu)
cat('\n*vertices ',n,'\n',file=clu)
for(i in 1:length(sx)) cat(' ',i,sx[i],file=sex)
cat('\n*vertices ',n,'\n',file=sex)
for(v in 1:n) {
  cat(v,' ',nodes$name[v],'\n',sep='',file=net);
  cat(M[v],'\n',file=clu); cat(S[v],'\n',file=sex)
}
for(r in 1:length(rel)) cat('*arcs :',r,' ',rel[r],'\n',sep='',file=net)
cat('*arcs\n',file=net)
for(a in 1:nrow(links))
  cat(R[a],': ',F[a],', ',T[a],', 1 1 ',rel[R[a]],'\n',sep='',file=net)
close(net); close(clu); close(sex)
```

CSV2Pajek.R



Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

```
*vertices 16
1 "Batagelj, Vladimir"
2 "Doreian, Patrick"
3 "Ferligoj, Anuška"
4 "Granovetter, Mark"
5 "Moustaki, Irini"
6 "Mrvar, Andrej"
7 "Clustering with relational constraint"
8 "The Strength of Weak Ties"
9 "Partitioning signed social networks"
10 "Generalized Blockmodeling"
11 "Psychometrika"
12 "Social Networks"
13 "The American Journal of Sociology"
14 "Structural Analysis in the Social Sciences"
15 "Cambridge University Press"
16 "Springer"
*arcs :1 "authorOf"
*arcs :2 "cites"
*arcs :3 "containedIn"
*arcs :4 "editorOf"
*arcs :5 "publishedBy"
```

```
*arcs
1: 1 10 1 1 "authorOf"
1: 2 10 1 1 "authorOf"
1: 3 10 1 1 "authorOf"
1: 1 7 1 1 "authorOf"
1: 3 7 1 1 "authorOf"
1: 4 8 1 1 "authorOf"
4: 4 14 1 1 "editorOf"
1: 2 9 1 1 "authorOf"
1: 6 9 1 1 "authorOf"
4: 5 11 1 1 "editorOf"
4: 2 12 1 1 "editorOf"
3: 10 14 1 1 "containedIn"
3: 7 11 1 1 "containedIn"
3: 8 13 1 1 "containedIn"
3: 9 12 1 1 "containedIn"
2: 9 10 1 1 "cites"
2: 10 7 1 1 "cites"
5: 14 15 1 1 "publishedBy"
5: 11 16 1 1 "publishedBy"
```

bib.net, bibMode.clu, bibSex.clu; bib.paj, bib.ini.



Bibliographic network – picture / Pajek

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

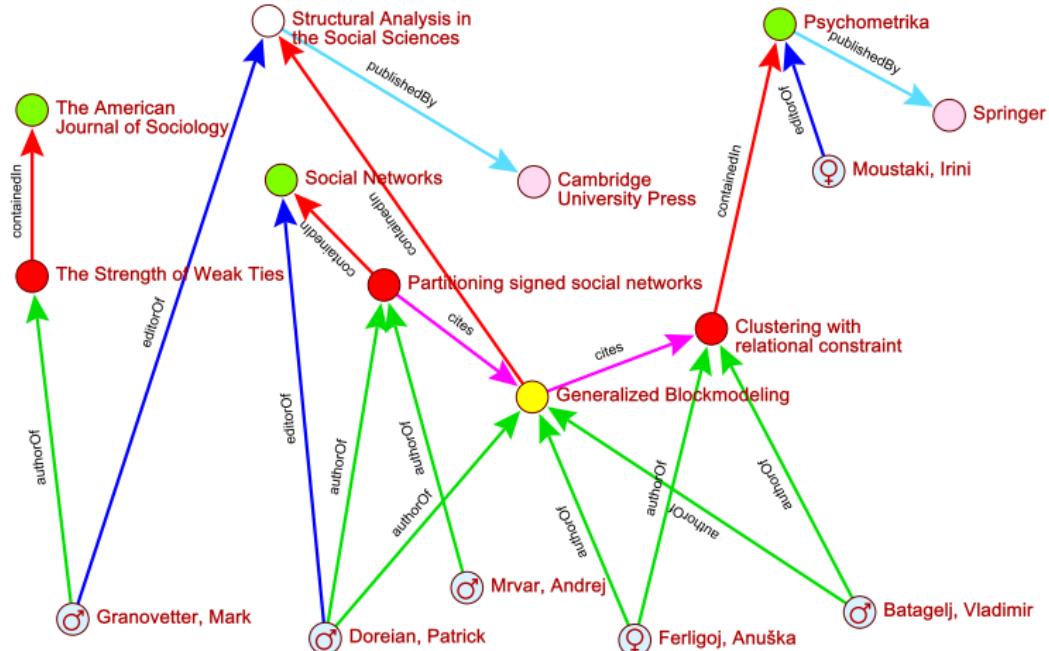
Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets





Reading Pajek files in R

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

wiki



Temporal network data

netsJSON format

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

For describing temporal networks we initially, extending Pajek format, defined and used a Janus format.

Recently we started to develop a new format based on JSON – we named it netsJSON (see [EDA: Data on files](#), slides 46-57).

netsJSON has two formats: a *basic* and a *general* format. Current implementation of the TQ library supports only the basic format. netsJSON format is supported by a Python library [Nets](#).



Informal description of the basic netJSON format

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netJSON
and Nets

```
{  
  "netsJSON": "basic",  
  "info": {  
    "org":1, "nNodes":n, "nArcs":mA, "nEdges":mE,  
    "simple":TF, "directed":TF, "multirel":TF, "mode":m,  
    "network":fName, "title":title,  
    "time": { "Tmin":tm, "Tmax":tM, "Tlabs": {labs} },  
    "meta": [events], ...  
  },  
  "nodes": [  
    { "id":nodeId, "lab":label, "x":x, "y":y, ... },  
    ***  
  ]  
  "links": [  
    { "type":arc/edge, "n1":nodeID1, "n2":nodeID2, "rel":r },  
    ***  
  ]  
}
```

where ... are user-defined properties and *** is a sequence of such elements.



Basic netsJSON format

Description

V. Batagelj

Networks

Descriptions
of networks

Properties

Types of
networks

Temporal
networks

Multi-
relational
networks

Two-mode
networks

igraph in R

Pajek and R

netsJSON
and Nets

An event description can contain fields:

```
{  "date": date,  
  "title": short description,  
  "author": name,  
  "desc": long description,  
  "url": URL,  
  "cite": reference,  
  "copy": copyright  
}
```

for describing temporal networks a node element and a link element has an additional required property `tq`

Example 1, Franzosi's violence network / UTF-8 no sig



Sources

V. Batagelj

How to get a
network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

Introduction to Network Analysis using **Pajek**

2. Sources of networks

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

PhD and MS program in Statistics
University of Ljubljana, 2022



Outline

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

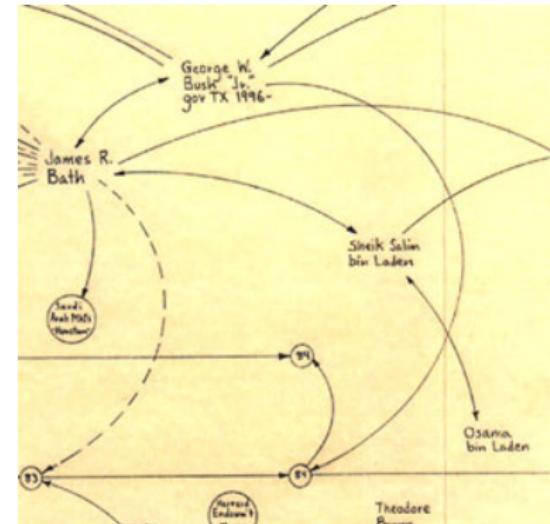
Neighbors

Transformations

Internet

Random

- 1 How to get a network?
- 2 Network data
- 3 GraphML
- 4 CaTA
- 5 Neighbors
- 6 Transformations
- 7 Internet
- 8 Random



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

Current version of slides (February 17, 2022 at 02 : 19): [slides PDF](#)



How to get a network?

Sources

V. Batagelj

How to get a
network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

Collecting data about the network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ we have first to decide, what are the units (nodes) – *network boundaries*, when are two units related – *network completeness*, and which properties of nodes/links we shall consider.

How to measure networks (questionnaires, interviews, observations, archive records, experiments, . . .)?

What is the quality of measured networks (reliability and validity)?
Privacy issues!

Several networks are already available in computer readable form or can be constructed from such data.

For large sets of units we often can't measure the complete network. Therefore we limit the data collection to selected units and their neighbors. We get *ego-centered networks*.



Use of existing network data

Sources

V. Batagelj

How to get a
network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

Pajek supports input of network data in several formats:
UCINET's DL files, graphs from project Vega, molecules in
MDLMOL, MAC, BS; genealogies in GEDCOM.

[Davis.DAT](#), [C84N24.VGR](#), MDL, [1CRN.BS](#), [DNA.BS](#),
[ADF073.MAC](#), [Bouchard.GED](#).

Several network data sets are already available in computer
readable form and need only to be transformed into network
descriptions.

[Wikipedia](#), [Internet Movie Data Base](#), [Digital Bibliography & Library Project](#), [CiteSeer](#), ...

For transformation of textual (tabular) data into **Pajek**'s
network the Jürgen Pfeffer's [txt2pajek](#) can be useful.



Krebs Internet industries

Sources

V. Batagelj

How to get a network?

Network data

GraphML

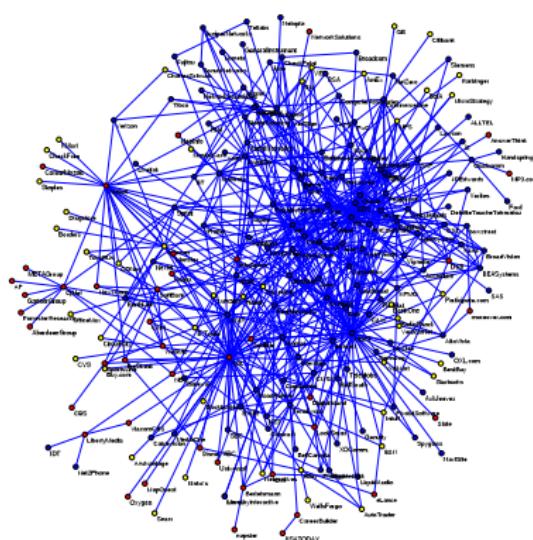
CaTA

Neighbors

Transformations

Internet

Random



Each node in the network represents a company that competes in the Internet industry, 1998 do 2001.

$n = 219$, $m = 631$.

red – content,

blue – infrastructure,

yellow – commerce.

Two companies are connected with an edge if they have announced a joint venture, strategic alliance or other partnership.

URL: <http://www.orgnet.com/netindustry.html>.
Recode, InfoRapid.



Genealogies

Sources

V. Batagelj

How to get a
network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

For describing the genealogies on computer most often the GEDCOM format is used (*GEDCOM standard 5.5*).

Many such genealogies (files *.GED) can be found on the Web – for example *Roper's GEDCOMs* or *Isle-of-Man GEDCOMs*. For scientific genealogies see *Kinsources*.

Several programs are available for preparation and maintainance of genealogies – for example *Brothers Keeper*.

From the data collected in Phd. thesis:

Mahnken, Irmgard. 1960. Dubrovački patricijat u XIV veku.
Beograd, Naučno delo.

the *Ragusa* network was produced.



GEDCOM

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

GEDCOM is a standard for storing genealogical data, which is used to interchange and combine data from different programs, which were used for entering the data.

```
0 HEAD
1 FILE ROYALS.GED
...
0 @I158@ INDI
1 NAME Charles Philip Arthur/Windsor/
1 TITL Prince
1 SEX M
1 BIRT
2 DATE 14 NOV 1948
2 PLAC Buckingham Palace, London
1 CHR
2 DATE 15 DEC 1948
2 PLAC Buckingham Palace, Music Room
1 FAMS @F16@
1 FAMC @F140
...
0 @I165@ INDI
1 NAME Diana Frances /Spencer/
1 TITL Lady
1 SEX F
1 BIRT
2 DATE 1 JUL 1961
2 PLAC Park House, Sandringham
1 CHR
2 PLAC Sandringham, Church
1 FAMS @F16@
1 FAMC @F78@
...
0 @I115@ INDI
1 NAME William Arthur Philip/Windsor/
1 TITL Prince
1 SEX M
1 BIRT
2 DATE 21 JUN 1982
2 PLAC St.Mary's Hospital, Paddington
1 CHR
2 DATE 4 AUG 1982
2 PLAC Music Room, Buckingham Palace
1 FAMC @F16@
...
0 @I116@ INDI
1 NAME Henry Charles Albert/Windsor/
1 TITL Prince
1 SEX M
1 BIRT
2 DATE 15 SEP 1984
2 PLAC St.Mary's Hosp., Paddington
1 FAMC @F16@
...
0 @F16@ FAM
1 HUSB @I158@
1 WIFE @I165@
1 CHIL @I115@
1 CHIL @I116@
1 DIV N
1 MARR
2 DATE 29 JUL 1981
2 PLAC St.Paul's Cathedral, London
```



Network representations of genealogies

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

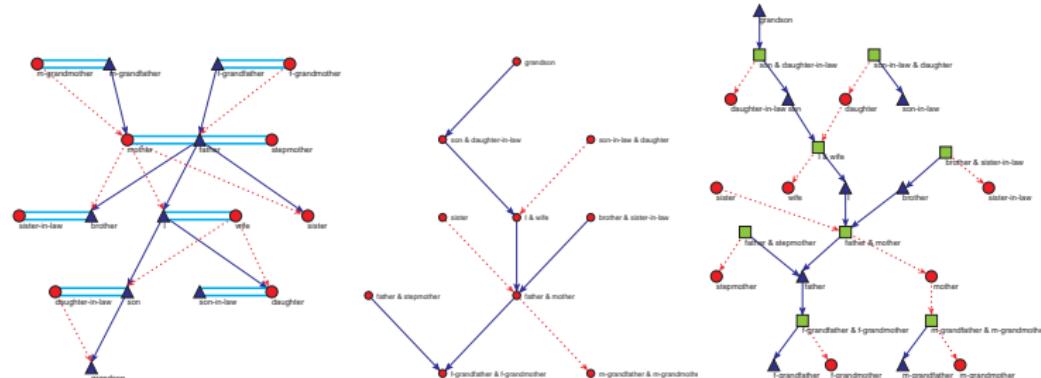
Neighbors

Transformations

Internet

Random

In a usual *Ore* graph every person is represented with a node; they are linked with two relations: *are married* (blue edge) and *has child* (black arc) – partitioned into *is mother of* and *is father of*. In a *p-graph* the nodes are married couples or singles; they are linked with two relations: *is son of* (solid blue) and *is daughter of* (dotted red). More about p-graphs *D. White*.



Ore graph, p-graph, and bipartite p-graph



Molecular networks

Sources

V. Batagelj

How to get a network?

Network data

GraphML

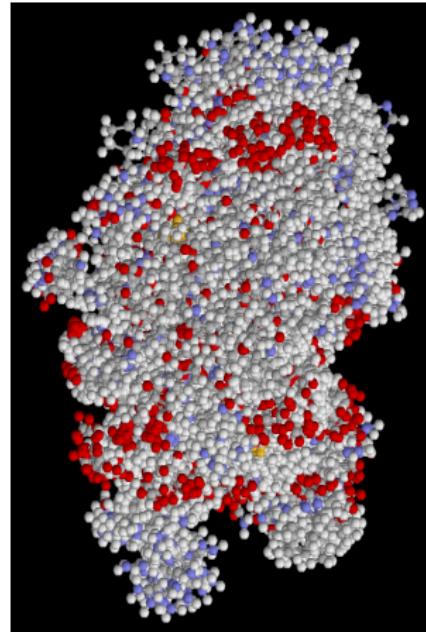
CaTA

Neighbors

Transformations

Internet

Random



virus 1G DY: $n = 39865$, $m = 40358$

In the [Brookhaven Protein Data Bank](#) we can find many large organic molecules (for example: Simian / 1AZ5.pdb) stored in PDB format.

They can be inspected in 3D using the program [Rasmol](#) (*RasMol*, *program*, *RasWin*) or [Protein Explorer](#).

A molecule can be converted from PDB format into BS format (supported by [Pajek](#)) using the program [BabelWin](#) + [Babel16](#).



GraphML

Sources

V. Batagelj

How to get a
network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

GraphML – XML format for network description.
L’Institut de Linguistique et Phonétique Générales et
Appliquées (ILPGA), Paris III; Traitement Automatique du
Langage (TAL): **BaO4 : Des Textes Aux Graphes Plurital**
LibXML, xsltproc download, XSLT, Xalan, Python, Sxslt.

```
xsltproc GraphML2Pajek.xsl graph.xml > graph.net
java -jar saxon8.jar graph.xml GraphML2Pajek.xsl > graph.
java org.apache.xalan.xslt.Process -IN p.xml -XSL m.xsl -
```

XSLT/Zvon

GraphML → Pajek

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Title: 1. D:\vlado\docs\Books\SKRIPTA\Nets\nets\graph.net (12) -->
<!-- Creator: Pajek: http://vlado.fmf.uni-lj.si/pub/networks/pajek/ -->
<!-- CreationDate: 11-03-2006, 17:25:13 -->
<graphml>
    <key id="a1" for="node" attr.name="Label" attr.type="string">
        <desc>Label of the node</desc> <default>NoLabel</default>
    </key>
    <key id="b1" for="edge" attr.name="Weight" attr.type="double">
        <desc>Weight (value) of the edge</desc> <default>1</default>
    </key>
    <graph id="G" edgedefault="directed" parse.nodes="12" parse.edges="23">
        <node id="v1"><data key="a1">a</data></node>
        <node id="v2"><data key="a1">b</data></node>
        <node id="v3"><data key="a1">c</data></node>
        <node id="v4"><data key="a1">d</data></node>
        <node id="v5"><data key="a1">e</data></node>
        <node id="v6"><data key="a1">f</data></node>
        <node id="v7"><data key="a1">g</data></node>
        <node id="v8"><data key="a1">h</data></node>
        <node id="v9"><data key="a1">i</data></node>
        <node id="v10"><data key="a1">j</data></node>
        <node id="v11"><data key="a1">k</data></node>
        <node id="v12"><data key="a1">l</data></node>
        <edge source="v1" target="v2"/> <edge source="v2" target="v1"/>
        <edge source="v1" target="v4"/> <edge source="v1" target="v6"/>
        <edge source="v2" target="v6"/> <edge source="v3" target="v2"/>
        <edge source="v3" target="v3"/> <edge source="v3" target="v7"/>
        <edge source="v3" target="v7"/> <edge source="v5" target="v3"/>
        <edge source="v5" target="v6"/> <edge source="v5" target="v8"/>
        <edge source="v6" target="v11"/> <edge source="v8" target="v4"/>
        <edge source="v10" target="v8"/> <edge source="v12" target="v5"/>
        <edge source="v12" target="v7"/> <edge source="v8" target="v12"/>
        <edge source="v12" target="v8"/>
        <edge directed="false" source="v2" target="v5"/>
        <edge directed="false" source="v3" target="v4"/>
        <edge directed="false" source="v5" target="v7"/>
        <edge directed="false" source="v6" target="v8"/>
    </graph>
</graphml>
```

*Vertices
1 "a"
2 "b"
3 "c"
4 "d"
5 "e"
6 "f"
7 "g"
8 "h"
9 "i"
10 "j"
11 "k"
12 "l"
*Edges
2 5
3 4
5 7
6 8
*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

GraphML → Pajek

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="text" encoding="iso-8859-1"/>
  <xsl:template match="/">
    <xsl:text>*Vertices </xsl:text>
    <xsl:value-of select="count(graphml/graph/node)"/>
    <xsl:text>&#10;.</xsl:text>
    <xsl:apply-templates select="graphml/graph/node"/>
    <xsl:text>*Edges&#10;.</xsl:text>
    <xsl:apply-templates select="graphml/graph/edge" mode="edge"/>
    <xsl:text>*Arcs&#10;.</xsl:text>
    <xsl:apply-templates select="graphml/graph/edge" mode="arc"/>
  </xsl:template>

  <xsl:template match="edge" mode="arc">
    <xsl:if test="not(@directed='false')">
      <xsl:value-of select="substring(./@source,2)"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="substring(./@target,2)"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select=".//data"/>
      <xsl:text>&#10;.</xsl:text>
    </xsl:if>
  </xsl:template>

  <xsl:template match="edge" mode="edge">
    <xsl:if test=".//@directed='false' ">
      <xsl:value-of select="substring(./@source,2)"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="substring(./@target,2)"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select=".//data"/>
      <xsl:text>&#10;.</xsl:text>
    </xsl:if>
  </xsl:template>

  <xsl:template match="node">
    <xsl:value-of select="substring(./@id,2)"/>
    <xsl:text> "</xsl:text>
    <xsl:value-of select=".//data"/>
    <xsl:text>"&#10;.</xsl:text>
  </xsl:template>
</xsl:stylesheet>
```



Computer-assisted text analysis

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

An often used way to obtain networks is the *computer-assisted text analysis* (CaTA).

Terms considered in TA are collected in a *dictionary* (it can be fixed in advance, or built dynamically). The main two problems with terms are *equivalence* (different words representing the same term) and *ambiguity* (same word representing different terms). Because of these the *coding* – transformation of raw text data into formal *description* – is done often manually or semiautomatically. As *units* of TA we usually consider clauses, statements, paragraphs, news, messages, . . .

Solutions for names: ResearcherID, ORCID, AMS; for words: dictionaries, stemming, lemmatization.

Till now the thematic and semantic TA mainly used statistical methods for analysis of the coded data.



... approaches to CaTA

Sources

V. Batagelj

How to get a
network?

Network data

GraphML

CaTA

Neighbors

Transformations

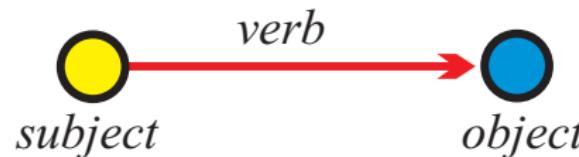
Internet

Random

In thematic TA the units are coded as rectangular matrix
Text units \times *Concepts* which can be considered as a two-mode network.

Examples: M.M. Miller: VBPro, H. Klein: Text Analysis/
TextQuest.

In semantic TA the units (often clauses) are encoded according to the S-V-O (*Subject-Verb-Object*) model or its improvements.



Examples: Roberto Franzosi; KEDS, Tabari, KEDS / Gulf.

This coding can be directly considered as network with *Subjects* \cup *Objects* as nodes and links labeled with *Verbs*.

See also RDF triples in semantic web, SPARQL.



Network CaTA

Sources

V. Batagelj

How to get a network?

Network data

GraphML

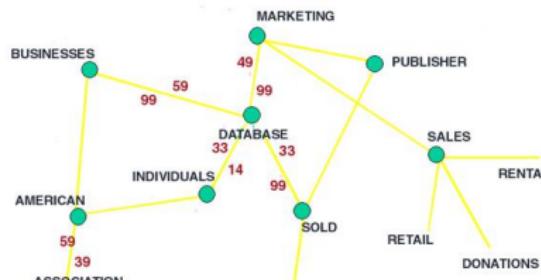
CaTA

Neighbors

Transformations

Internet

Random



TextAnalyst's 'semantic network'

This way we already stepped into the network TA.

Examples:

Carley: Cognitive maps,
J.A. de Ridder: CETA,
Megaputer: TextAnalyst.

See also: W. Evans: Computer Environments for Content Analysis, K.A. Neuendorf: The Content Analysis Guidebook / Online and H.D. White: Publications.

There are additional ways to obtain networks from textual data.



TA – International Relations

Sources

V. Batagelj

How to get a
network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

*Paul Hensel's International Relations Data Site,
International Conflict and Cooperation Data,
Correlates of War,
Kansas Event Data System KEDS,
KEDS in Pajek's format.
Recoding programs in R.*



Multi-relational temporal network – KEDS/WEIS

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

```
% 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 ET
224: 3 83 1 [4] 890407 ALB ETHALB 224 (RIOT) RIOTS
123: 83 153 1 [4] 890408 ETHALB KSV 123 (INVESTIGATE) PROBIN

...
42: 105 63 1 [175] 030731 GER CYP 042 (ENDORSE) GAVE S
212: 295 35 1 [175] 030731 UNWCT BOSSER 212 (ARREST PERSON) SENTEN
43: 306 87 1 [175] 030731 VAT EUR 043 (RALLY) RALLIED
13: 295 35 1 [175] 030731 UNWCT BOSSER 013 (RETRACT) CLEARE
121: 295 22 1 [175] 030731 UNWCT BAL 121 (CRITICIZE) CHARGE
122: 246 295 1 [175] 030731 SER UNWCT 122 (DENIGRATE) TESTIF
121: 35 295 1 [175] 030731 BOSSER UNWCT 121 (CRITICIZE) ACCUSE
```



... Program in R

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

```
# WEISmonths
# recoding of WEIS files into Pajek's multirelational temporal files
# granularity is 1 month
#
# Vladimir Batagelj, 28. November 2004
#
# Usage:
#   WEISmonths(WEIS_file,Pajek_file)
# Examples:
#   WEISmonths('Balkan.dat','BalkanMonths.net')
#
# http://www.ku.edu/~keds/data.html
#
WEISmonths <- function(fdat,fnet){
  get.codes <- function(line){
    nlin <- nlin + 1;
    z <- unlist(strsplit(line,"\\t"));
    z <- z[z != ""]
    if (length(z)>4) {
      t <- as.numeric(z[1]);
      if (t < 500000) t <- t + 1000000
      if (t<t0) t0 <- t; u <- z[2]; v <- z[3]; r <- z[4];
      if (is.na(as.numeric(r))) cat(nlin,'NA rel-code',r,'\n')
      h <- z[5]; h <- substr(h,2,nchar(h)-1)
      if (nchar(h) == 0) h <- "*** missing description"
      if (!exists(u,env=act,inherits=FALSE)){
        nver <- nver + 1; assign(u,nver,env=act) }
      if (!exists(v,env=act,inherits=FALSE)){
        nver <- nver + 1; assign(v,nver,env=act) }
      if (!exists(r,env=rel,inherits=FALSE)) assign(r,h,env=rel)
    }
  }
}
```

... Program in R

```
recode <- function(line){  
  nlin <- nlin + 1;  
  z <- unlist(strsplit(line, "\t")); z <- z[z != ""]  
  if (length(z)>4) {  
    t <- as.numeric(z[1]); if (t < 500000) t <- t + 1000000  
    cat(as.numeric(z[4]),':',get(z[2],env=act,inherits=FALSE),  
        ' ',get(z[3],env=act,inherits=FALSE),' 1 [',  
        12*(1900 + t %% 10000) + (t %% 10000) %% 100 - t0,  
        ']\n',sep='',file=net)  
  }  
}  
  
cat('WEISmonths: WEIS -> Pajek\n')  
ts <- strsplit(as.character(Sys.time())," ")[[1]][2]  
act <- new.env(TRUE,NULL); rel <- new.env(TRUE,NULL)  
dat <- file(fdat,"r"); net <- file(fnet,"w")  
lst <- file('WEIS.lst',"w"); dni <- 0  
nver <- 0; nlin <- 0; t0 <- 9999999  
lines <- readLines(dat); close(dat)  
sapply(lines,get.codes)  
a <- sort(ls(envir=act)); n <- length(a)  
cat(paste('% Recoded by WEISmonths,',date(),"\n",file=net)  
cat("% from http://www.ku.edu/~keds/data.html\n",file=net)  
cat("*vertices",n," \n",file=net)  
for(i in 1:n){ assign(a[i],i,env=act);  
  cat(i,' ',a[i]," [--]\n",sep='',file=net) }  
b <- sort(ls(envir=rel)); m <- length(b)  
for(i in 1:m){ assign(a[i],i,env=act);  
  cat("*arcs :",as.numeric(b[i]),' ','\n',sep='',file=net) }  
t0 <- 12*(1900 + t0 %% 10000)  
slice <- 0  
cat("*arcs\n",file=net); nlin <- 0  
sapply(lines,recode)  
cat(' ',nlin,'lines processed\n'); close(net)  
te <- strsplit(as.character(Sys.time())," ")[[1]][2]  
cat(' start:',ts,' finish:',te,'\n')  
}  
  
WEISmonths('Balkan.dat','BalkanMonthsR.net')  
Note: In R to a dictionary data structure corresponds the notion of environment.
```



Dictionary networks

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

book

A collection of leaves of [paper](#), [parchment](#), [vellum](#), cloth, or other material (written, [printed](#), or [blank](#)) fastened together along one edge, with or without a protective [case](#) or [cover](#). Also refers to a literary [work](#) or one of its [volumes](#). Compare with [monograph](#).

To qualify for the special parcel post rate known in the United States as [media rate](#), a [publication](#) must consist of 24 or more [pages](#), at least 22 of which bear [printing](#) consisting primarily of reading material or scholarly [bibliography](#), with advertising limited to [book announcements](#). UNESCO defines a book as a non[periodical](#) literary publication consisting of 49 or more pages, covers excluded. The [ANSI standard](#) includes publications of less than 49 pages which have [hard covers](#). See also: [art book](#), [board book](#), [children's book](#), [coffee table book](#), [gift book](#), [licensed book](#), [managed book](#), [new book](#), [packaged book](#), [picture book](#), [premium book](#), [professional book](#), [promotional book](#), [rare book](#), [reference book](#), [religious book](#), and [reprint book](#).

Also, a major division of a longer [work](#) (usually of [fiction](#)) which is further subdivided into [chapters](#). Usually [numbered](#), such a division may or may not have its own [title](#). Also refers to one of the divisions of the Christian [Bible](#), the first being [Genesis](#).

[book](#) description in ODLIS

The Edinburgh Associative Thesaurus ([EAT](#)) / [net](#); NASA Thesaurus.

In a *dictionary graph* the terms determine the set of nodes, and there is an arc (u, v) from term u to term v iff the term v appears in the description of term u .

Online Dictionary of Library and Information Science

[ODLIS](#), [Odlis.net](#) (2909 / 18419).

Free On-line Dictionary of Computing [FOLDOC](#), [Foldoc2b.net](#) (133356 / 120238).

[Artlex](#), [Wordnet](#), [Concept-Net](#), [OpenCyc](#).



Collaboration networks

Sources

V. Batagelj

How to get a
network?

Network data

GraphML

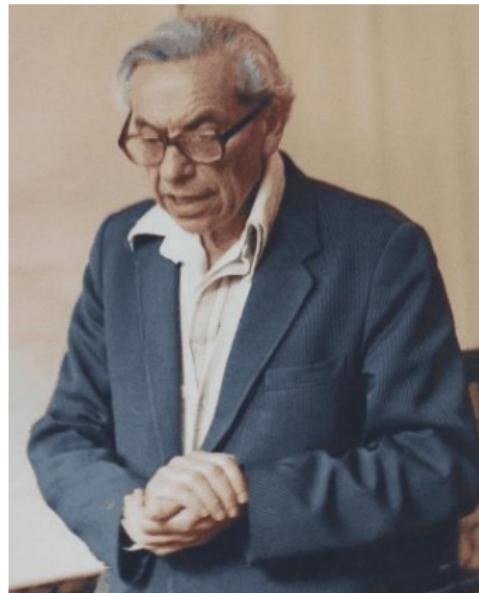
CaTA

Neighbors

Transformations

Internet

Random



Units in a *collaboration network* are usually individuals or institutions. Two units are related if they produced a joint work. The weight is the number of such works. A famous example of collaboration network is *The Erdős Number Project, Erdos.net*.

A rich source of data for producing collaboration networks are the BibTeX bibliographies *Nelson H. F. Beebe's Bibliographies Page*.

For example B. Jones: *Computational geometry database* (2002), *FTP, Geom.net*.

An initial collaboration network from such data can be produced using some programming. Then follows a tedious 'cleaning' process.

Interesting datasets: *The Internet Movie Database* and *Trier DBLP*.

Both citation and collaboration networks can be obtained from *Web of Science* using *WoS2Pajek*. See also *Bibexcel*.



Neighbors

Sources

V. Batagelj

How to get a
network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

Let \mathcal{V} be a *set of multivariate units* and $d(u, v)$ a *dissimilarity* on it.
They determine two types of networks:

The *k-nearest neighbors* network: $\mathcal{N}(k) = (\mathcal{V}, \mathcal{A}, d)$

$(u, v) \in \mathcal{A} \Leftrightarrow v$ is among k nearest neighbors of u , $w(u, v) = d(u, v)$

The *r-neighbors* network: $\mathcal{N}(r) = (\mathcal{V}, \mathcal{E}, d)$

$(u : v) \in \mathcal{E} \Leftrightarrow d(u, v) \leq r$, $w(u, v) = w(v, u) = d(u, v)$

These networks provide a link between data analysis and network analysis. Efficient algorithms ?!
Nearest neighbor library in R-package *yamlpute*.

Fisher's *Iris data*. Details on *Multivariate networks* and procedures in R.



Nearest k neighbors in R

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

```
k.neighbor2Net <-
# stores network of first k neighbors for
# dissimilarity matrix d to file fnet in Pajek format.
function(fnet,d,k){
  net <- file(fnet,"w")
  n <- nrow(d); rn <- rownames(d)
  cat("*vertices",n,"\n",file=net)
  for (i in 1:n) cat(i," \",rn[i],"\\"\\n",sep="",file=net)
  cat("*arcs\\n",file=net)
  for (i in 1:n) for (j in order(d[i,])[1:k+1]) {
    cat(i,j,d[i,j],"\\n",file=net)
  }
  close(net)
}

data(iris)
ir <- scale(iris)
rownames(ir) <- paste(substr(iris[,5],1,2),1:nrow(iris),sep="")
k.neighbor2Net("iris5.net",as.matrix(dist(ir)),5)
```



Fast nearest k neighbors in R

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

David M. Mount wrote the Approximate Nearest Neighbor Library (<http://www.cs.umd.edu/~mount/ANN>) with fast algorithms for the (approximate) nearest neighbor search. In R these algorithms are available through function `ann` in package `yaImpute`.

```
k.neighbor2NetF <-
# stores network of first k neighbors for data matrix d to file fnet
# in Pajek format.
# Example:
#   data(iris); stand <- function(x) {(x-mean(x))/sd(x)}
#   ir <- cbind(stand(iris[,1]),stand(iris[,2]),stand(iris[,3]),
#   #   stand(iris[,4]))
#   k.neighbor2NetF("iris5Y.net",ir,5)
# V. Batagelj, 8.8.2009 yaImpute / 9.9.2008 knnFinder
function(fnet,d,k){
  library(yaImpute)
  NN <- ann(ir,target=ir,k=k+1)
  net <- file(fnet,"w")
  n <- nrow(d)
  rn <- if (is.null(rownames(d))) paste("U-",1:n,sep='') else rownames(d)
  cat("*vertices",n,"\\n",file=net)
  for (i in 1:n) cat(i,"\"",rn[i], "\""\\n",sep="",file=net)
  cat("*arcs\\n",file=net)
  for (i in 1:n) for (j in 1:k)
    cat(i,NN$knnIndexDist[i,j+1],NN$knnIndexDist[i,j+k+2],"\\n",file=net)
  close(net)
}
```



Fisher's Irises

Sources

V. Batagelj

How to get a network?

Network data

GraphML

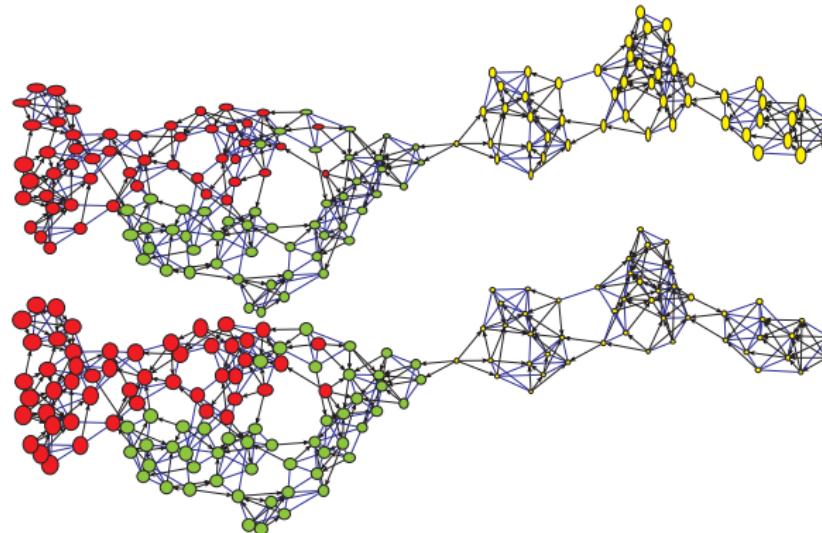
CaTA

Neighbors

Transformations

Internet

Random



Draw/Network+First Partition+First Vector+Second Vector

The size of nodes is proportional to normalized (Sepal.Length, Sepal.Width) and (Petal.Length, Petal.Width). The color of nodes is determined by the original partition. *Iris data.*



r-neighbors in R

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

```
r.neighbor2Net <-
# stores network of r-neighbors (d(v,u) <= r) for
# dissimilarity matrix d to file fnet in Pajek format.
function(fnet,d,r){
  net <- file(fnet,"w")
  n <- nrow(d); rn <- rownames(d)
  cat ("*vertices",n,"\n",file=net)
  for (i in 1:n) cat(i, " \",rn[i], "\n",sep="",file=net)
  cat ("*edges\n",file=net)
  for (i in 1:n){
    s <- order(d[i,]); j <- 1
    while (d[i,s[j]] <= r) {
      k <- s[j]; if (i < k) cat(i,k,d[i,k],"\n",file=net)
      j <- j+1
    }
  }
  close(net)
}
```



Transformations

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

Words graph – words from a given set are nodes; two words are related iff one can be obtained from the other by change (add, delete, replace) of a single character. [DIC28](#), [Paper](#).

Text network – nodes are (selected) words from a given text; two words are related if they coappeared in the selected type of 'window' (same sentence, k consecutive words, ...) The weights count such coappearances. Example [CRA](#).

Game graph – nodes are states in the game; two states are linked with an arc if the rules of the game allow the transition from first to the second state. [DMFA'08](#).

Using the information from mobile phones or RFIDs (Radio-frequency identification) the **networks of interactions** of their owners can be constructed.



Networks from the Internet

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random



KartOO network

Semantic web (URI, RDF, OWL). LOD, FreeBase, DBpedia.

Internet Mapping Project.
Links among WWW pages.
KartOO, TouchGraph.

Derived from archives of E-mail, blogs, ..., server's logs.

Cybergeography, CAIDA.

Tools: *MedlineR*, *SocSci-Bot*.



Collecting Networks from WWW

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

Web wrappers are special programs for collecting information from web pages – often returned in XML format.

Examples in R: [Titles of patents from Nber](#), [Books from Amazon](#).

Several tools for automatic generation of wrappers: ([paper](#) / [list](#) / [LAPIS](#)).

Free programs: XWRAP ([description](#) / [page](#)) in TSIMMIS ([description](#) / [page](#)).

Among commercial programs it seems the best is [lixto](#).

Additional URLs [1](#), [2](#), [3](#).

[Nutch](#), [IssueCrawler](#), [W4F](#).

Python: [lxml](#); [Beautiful Soup](#).

[Amazon web services](#), [Google Data](#), [Google+](#), [YouTube](#), [Twitter](#), [Last.fm](#), [MusicBrainz3](#), [Flickr](#), [LinkedIn](#), ...



Networks from Amazon in R

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

Amazon is changing the structure of pages. Probably this program doesn't work correctly.

```
amazon <- function(fvtx, flnk, ftit, maxver) {
  # Creates a network of books from Amazon
  #   amazon('v.txt','a.txt','t.txt',10)
  # Vladimir Batagelj, 20-21. nov. 2004 / 10. nov. 2006
  opis <- function(line){
    i <- regexpr('>',line); l <- i[1]+attr(i,"match.length")[-1]
    j <- regexpr('</a>',line); r <- j[1]-1; substr(line,l,r)
  }
  vid <- new.env(hash=TRUE,parent=emptyenv())
  vtx <- file(fvtx,"w"); cat('*vertices\n', file=vtx)
  tit <- file(ftit,"w"); cat('*vertices\n', file=tit)
  lnk <- file(flnk,"w"); cat('*arcs\n', file=lnk)
  url1 <- 'http://www.amazon.com/exec/obidos/tg/detail/-/'
  url2 <- '?v=glance';
  book <- '0521840856'
  auth <- "Patrick Doreian"
  titl <- "Generalized Blockmodeling"
  narc <- 0; nver <- 1
  page <- paste(url1,book,url2,sep='')
  cat(nver, ' ', book, ' URL "', page, '"\n', sep=' ', file=vtx)
  cat(nver, ' ', auth, ':\\n', titl, '\n', sep=' ', file=tit)
  assign(book,nver,env=vid)
  cat('new vertex',nver,' - ',book,'\n')
  books <- c(book)
```



... Networks from Amazon in R

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

```
while (length(books)>0){  
  bk <- books[1]; books <- books[-1]  
  vini <- get(bk,env=vid); cat(vini,'\n')  
  page <- paste(url1,bk,url2,sep='')  
  stran <- readLines(con<-url(page)); close(con)  
  i <- grep("Customers who bought",stran,ignore.case=TRUE)[1]  
  if (is.na(i)) break  
  j <- grep("Explore Similar Items",stran,ignore.case=TRUE)[1]  
  izrez <- stran[i:j]; izrez <- izrez[-which(izrez=="")]  
  izrez <- izrez[-which(izrez==" ")]  
  ik <- regexpr("/dp/",izrez); ii <- ik+attr(ik,"match.length")  
  for (k in 1:length(ii)) {  
    j <- ii[k];  
    if (j > 0) {  
      bk <- substr(izrez[k],j,j+9); cat('test',k,bk,'\n')  
      if (exists(bk,env=vid,inherits=FALSE)){  
        vter <- get(bk,env=vid,inherits=FALSE)  
      } else {  
        nver <- nver + 1; vter <- nver; line <- izrez[k]  
        assign(bk,nver,env=vid)  
        if (nver <= maxver) {books <- append(books,bk)}  
        cat(nver,' ',bk,'" URL "','"url1,bk,url2,"'\n',sep='',file=vtx)  
        cat('new vertex',nver,' - ',bk,'\n');  
        t <- opis(line); line <- izrez[k+1]  
        if (substr(line,1,2)=='by') (a <- substr(line,4,100))  
        else { a <- 'UNKNOWN' }  
        cat(nver, ' ', a, ':\\n', t, "'\\n', sep='', file=tit)  
      }  
      narc <- narc + 1; cat(vini,vter,'\n', file=lnk)  
    }  
  }  
  flush.console()  
}  
close(lnk); close(vtx); cat('Amazon - END\\n')
```



Networks from Amazon – books on SNA

Sources

V. Batagelj

How to get a network?

Network data

GraphML

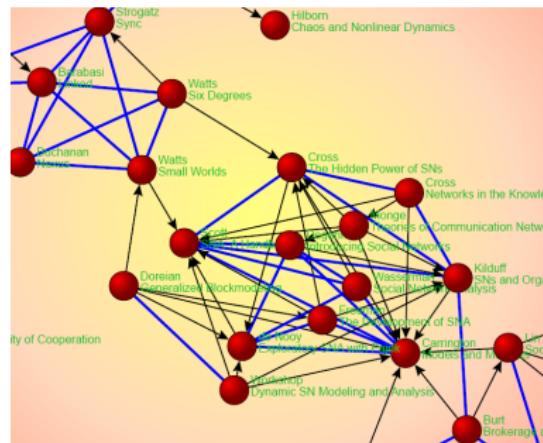
CaTA

Neighbors

Transformations

Internet

Random



Books in SNA from Amazon,
10. november 2006; Starting
point P. Doreian &: **General-
ized Blockmodeling**.

The program in R is just a
skeleton. Possible improve-
ments: list of starting points;
continuation after interrupts;
etc.

The structure of Amazon files
is changing!!!



Random networks

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

Several types of networks can be produced randomly using special generators. The theoretical **background** of these generators is beyond the goals of this course.

Some of them are implemented in **Pajek** under Network / Create Random Network but can be also described by the following **functions in R**.

Available is also a program **GeneoRnd** for generating random genealogies.

For generating random networks with special properties the **probabilistic inductive classes of graphs** can be used.



Random undirected graph of Erdős-Rényi type

Sources

V. Batagelj

How to get a network?

Network data

GraphML

CaTA

Neighbors

Transformations

Internet

Random

```
dice <- function(n=6){return(1+trunc(n*runif(1,0,1)))}

ErdosRenyiNet <-
# generates a random undirected graph of Erdos-Renyi type
# with n nodes and m edges, and stores it on the file
# fnet in Pajek's format.
# Example: ErdosRenyiNet('testER.net',100,175)
# -----
# by Vladimir Batagelj, R version: Ljubljana, 20. Dec 2004
# based on ALG.2 from: V. Batagelj, U. Brandes:
# Efficient generation of large random networks
function(fnet,n,m){
  net <- file(fnet,"w"); cat("*vertices",n,"\n",file=net)
  cat('% random Erdos-Renyi undirected graph G(n,m) / m = ',
      m,' \n',file=net)
#  for (i in 1:n) cat(i, " \\"v",i,"\\n",sep="",file=net)
  cat("*edges\n",file=net); L <- new.env(TRUE,NULL)
  for (i in 1:m){
    repeat { u <- dice(n); v <- dice(n)
      if (u!=v) {
        edge <- if (u<v) paste(u,v) else paste(v,u)
        if (!exists(edge,env=L,inherits=FALSE)) break }
      assign(edge,0,env=L); cat(edge,' \n',file=net)
    }
  close(net)
}
```



Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

Introduction to Network Analysis using **Pajek**

3. Structure of networks: subnetworks

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

PhD and MS program in Statistics
University of Ljubljana, 2022



Outline

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

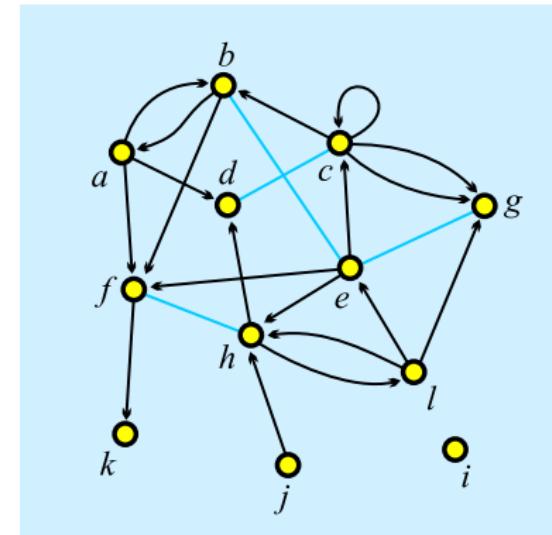
Morphisms

Partitions

Subgraphs

Cuts

- 1 Size of networks
- 2 Pajek
- 3 Statistics
- 4 Morphisms
- 5 Partitions
- 6 Subgraphs
- 7 Cuts



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

Current version of slides (February 17, 2022 at 02:29): [slides PDF](#)



Degrees

Subnetworks

V. Batagelj

Size of networks

Pajek

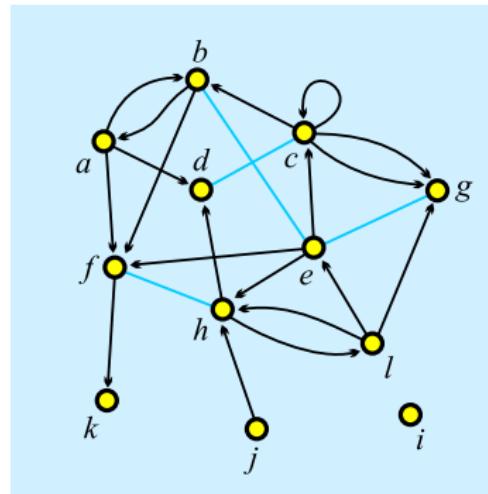
Statistics

Morphisms

Partitions

Subgraphs

Cuts



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

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

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

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

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

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

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



Size of network

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

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 details; special algorithms are needed for their analysis (**Pajek**).



Large networks

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

Large network – several thousands or millions of nodes. Can be stored in 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

Pajek datasets.



Dunbar's number

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

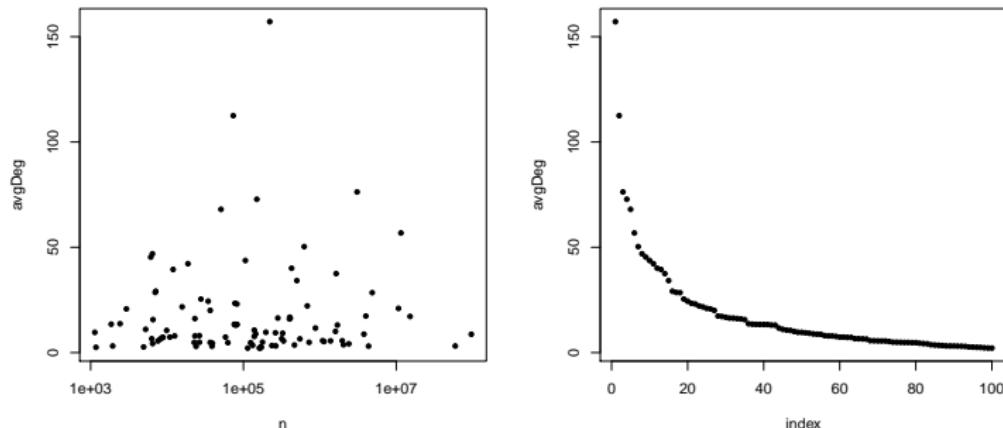
Morphisms

Partitions

Subgraphs

Cuts

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 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 the Dunbar's number is between 100 and 150.



Complexity of algorithms

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

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

Subnetworks

V. Batagelj

Size of networks

Pajek

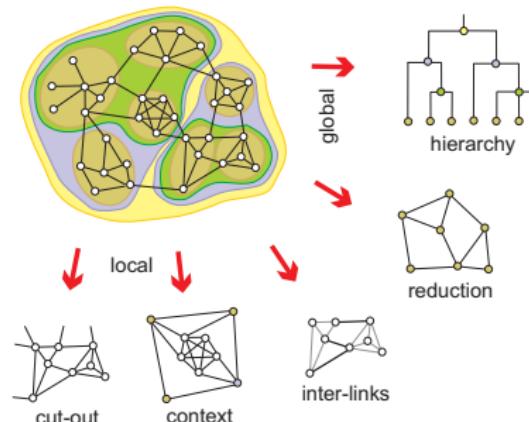
Statistics

Morphisms

Partitions

Subgraphs

Cuts



In 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 few algorithms available.

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



Pajek's data types

Subnetworks

V. Batagelj

Size of networks

Pajek

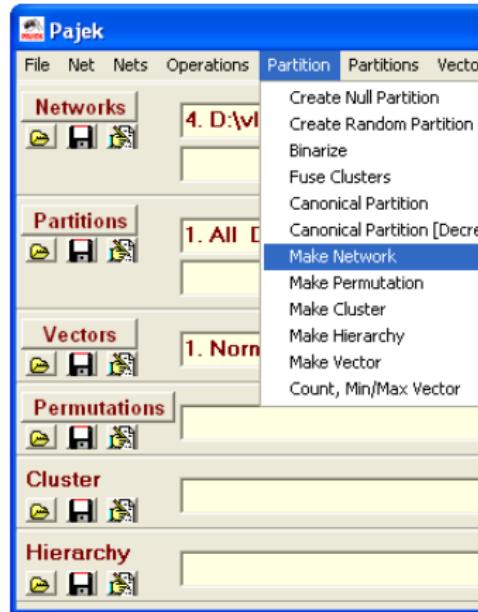
Statistics

Morphisms

Partitions

Subgraphs

Cuts



- *network* (graph),
- *partition* (nominal or ordinal properties of nodes),
- *vector* (numerical properties of nodes),
- *cluster* (subset of nodes),
- *permutation* (reordering of nodes, ordinal properties), and
- *hierarchy* (general tree structure on nodes).

Pajek supports also *multi-relational*, *temporal* and *two-mode* networks.



Pajek's data types

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

The power of **Pajek** is based on several transformations that support different transitions among these data structures. Also the menu structure of the main **Pajek**'s window is based on them. **Pajek**'s main window uses a 'calculator' paradigm with list-accumulator for each data type. The operations are performed on the currently active (selected) data and are also returning the results through accumulators.

The procedures are available through the main window menus. Frequently used sequences of operations can be defined as *macros*. This allows also the adaptations of **Pajek** to groups of users from different areas (social networks, chemistry, genealogy, computer science, mathematics...) for specific tasks. **Pajek** supports also *repetitive operations* on series of networks.



Statistics

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

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 (betweenness, hubs, authorities, ...)

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

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



... Statistics

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

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 association as collected from subjects (students). The weight on the arcs is the count of word associations.

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

It has 23219 nodes and 325624 arcs (564 loops); number of links with value=1 is 227481.



... Statistics

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

To identify the nodes with the largest degree:

Net/Partitions/Degree/All

Partition/Make vector

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

In igraph the function `degree()` has modes `in`, `out` and `all`.

```
> G <- read.graph("links.net", format="pajek")
> deg <- degree(G, mode="all")
> plot(G, vertex.size=deg*3)
```



Degrees in igraph

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

In the file **igraph+.R** some additional functions are collected that make network analysis easier. For example, the function **top**

```
top <- function(v,k) {
  ord <- rev(order(v)); sel <- ord[1:k]
  S <- data.frame(name=names(v[sel]),
    value=as.vector(v[sel]))
  return(S)
}
```

returns top k values in the node attribute v .

```
> wdir <- "C:/Users/batagelj/Documents/papers/2017/Moscow"
> setwd(wdir)
> library(igraph)
# delete *network and empty line before *vertices
> T <- read.graph("./nets/eatRS.net", format="pajek")
> vcount(T)
[1] 23219
> ecount(T)
[1] 325624
> source("igraph+.R")
> SR <- graph.reverse(T)
> SR$indeg <- degree(SR, mode="in")
```



... Degrees in igraph

Subnetworks

V. Batagelj

```
> top(SR$indeg,10)
      name value
1       ME   1074
2      MAN   1046
3     GOOD   861
4      SEX   828
5      NO    780
6    MONEY   743
7      YES   718
8     WORK   672
9    NOTHING   672
10    FOOD   665
> SR$windeg <- strength(SR, mode="in")
> max(SR$windeg)
[1] 4387
> top(SR$windeg,20)
> SR$awindeg <- SR$windeg/SR$indeg
> SR$awindeg[is.nan(SR$awindeg)] <- 0
> top(SR$awindeg,20)
```



Statistics / **Pajek** and R

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

Pajek (0.89 and higher) 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/Info/General

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 L^AT_EX; Windows metafile format for inclusion in Word).

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



EAT all-degree distribution

Subnetworks

V. Batagelj

Size of networks

Pajek

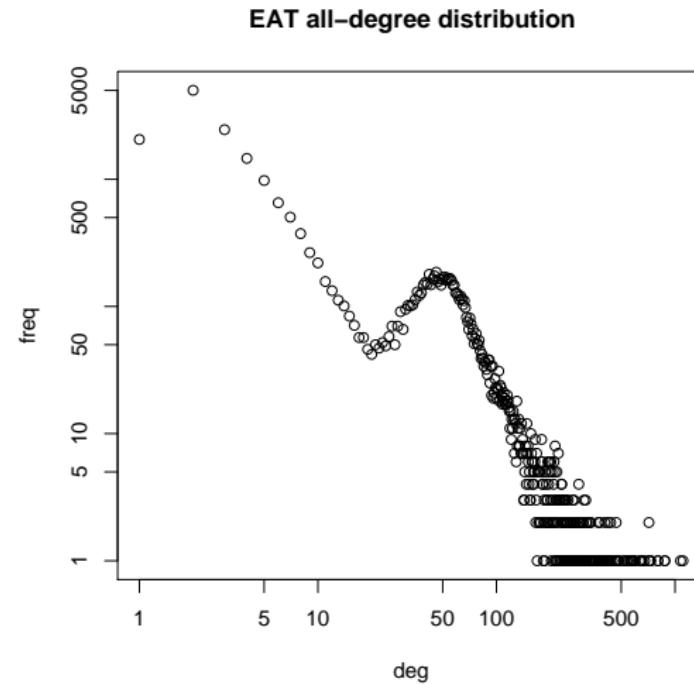
Statistics

Morphisms

Partitions

Subgraphs

Cuts





Erdős and Renyi's random graphs

Subnetworks

V. Batagelj

Size of networks

Pajek

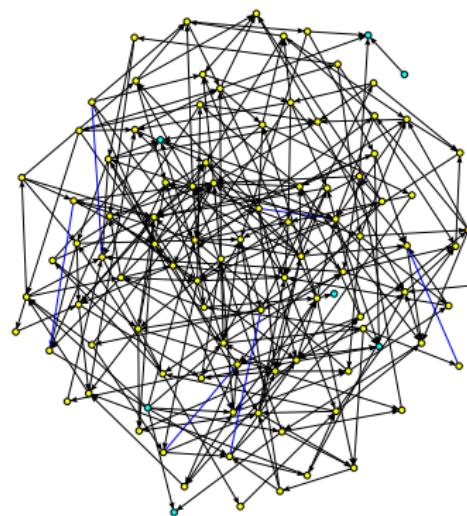
Statistics

Morphisms

Partitions

Subgraphs

Cuts



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

Network/Create

Random Network/

Bernoulli/Poisson/Undirected

General [100] [2.5]

instead of probability p a more intuitive average degree is used

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

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

Random graph in the picture has 100 nodes and average degree $\overline{\deg} = 2.5$.



Degree distribution

Subnetworks

V. Batagelj

Size of networks

Pajek

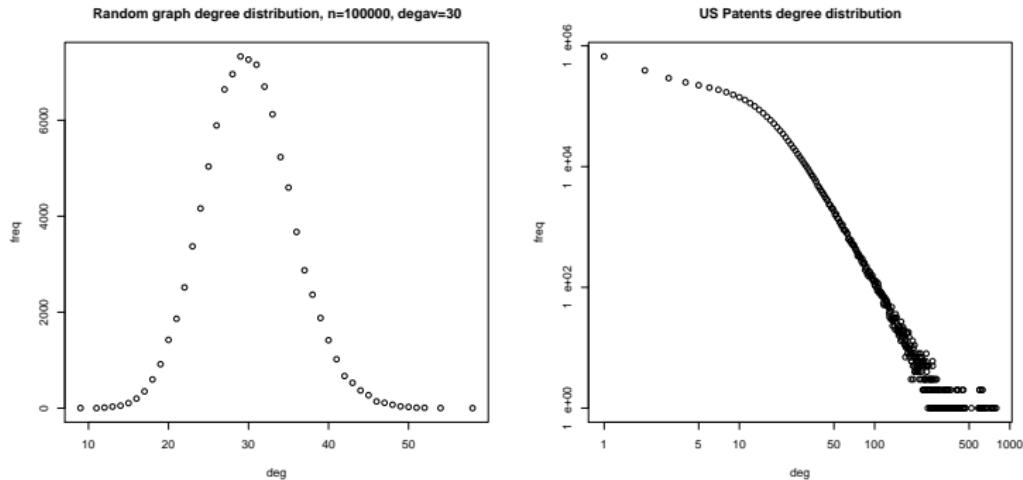
Statistics

Morphisms

Partitions

Subgraphs

Cuts



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

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

We read in **Pajek** the 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:
v3  : From partition 1 (548600)
v4  : From partition 2 (548600)
-----
> inTab <- table(v3)
> 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(v4)
> 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

Subnetworks

V. Batagelj

Size of networks

Pajek

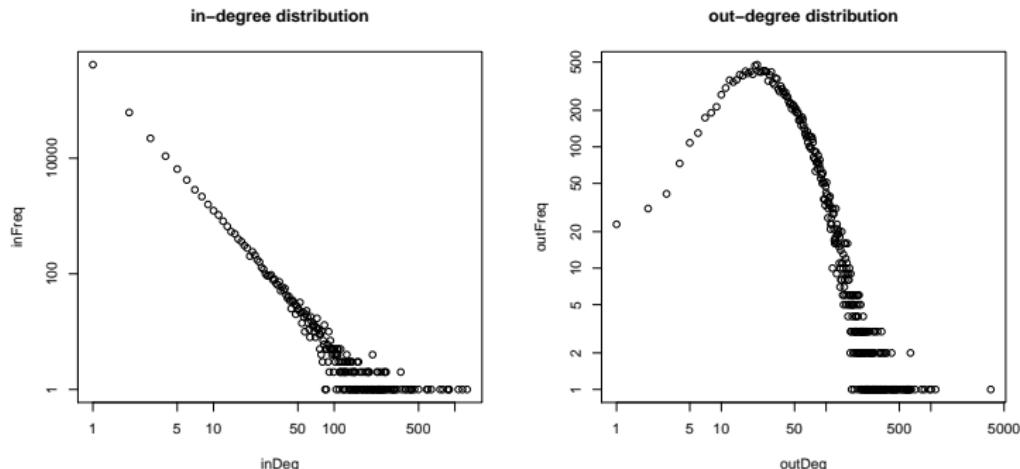
Statistics

Morphisms

Partitions

Subgraphs

Cuts



The in-degree distribution is "scale-free"-like. The parameters 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

Subnetworks

V. Batagelj

Size of networks

Pajek

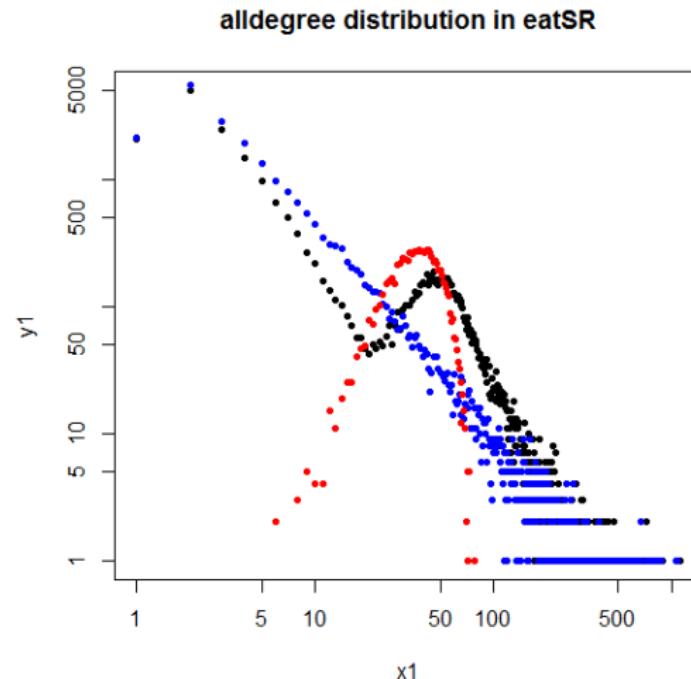
Statistics

Morphisms

Partitions

Subgraphs

Cuts





Papers by years / centrality network

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

From the file Year.clu containing the year of publication of a paper we can get the distribution of *papers by years*. For the centrality network we get:

```
> setwd("C:/Users/Batagelj/work/Python/WoS/Central")
> years <- read.table(file="Year.clu",header=FALSE,skip=2) $V1
> t <- table(years)
> year <- as.integer(names(t))
> freq <- as.vector(t[1950<=year & year<=2009])
> y <- 1950:2009
> plot(y,freq)
> model <- nls(freq~c*dlnorm(2010-y,a,b),
+ start=list(c=350000,a=2,b=0.7))
> model
Nonlinear regression model
  model: freq ~ c * dlnorm(2010 - y, a, b)
  data: parent.frame()
      c          a          b
 5.427e+05 2.491e+00 6.624e-01
 residual sum-of-squares: 20474181

Number of iterations to convergence: 7
Achieved convergence tolerance: 3.978e-06
> lines(y,predict(model,list(x=2010-y)),col='red')
```

It can be well approximated by the *lognormal distribution*, but also by the *generalized reciprocal power exponential curve* $c * (x + d)^{\frac{a}{b+x}}$.



Papers by years / centrality network

Subnetworks

V. Batagelj

Size of
networks

Pajek

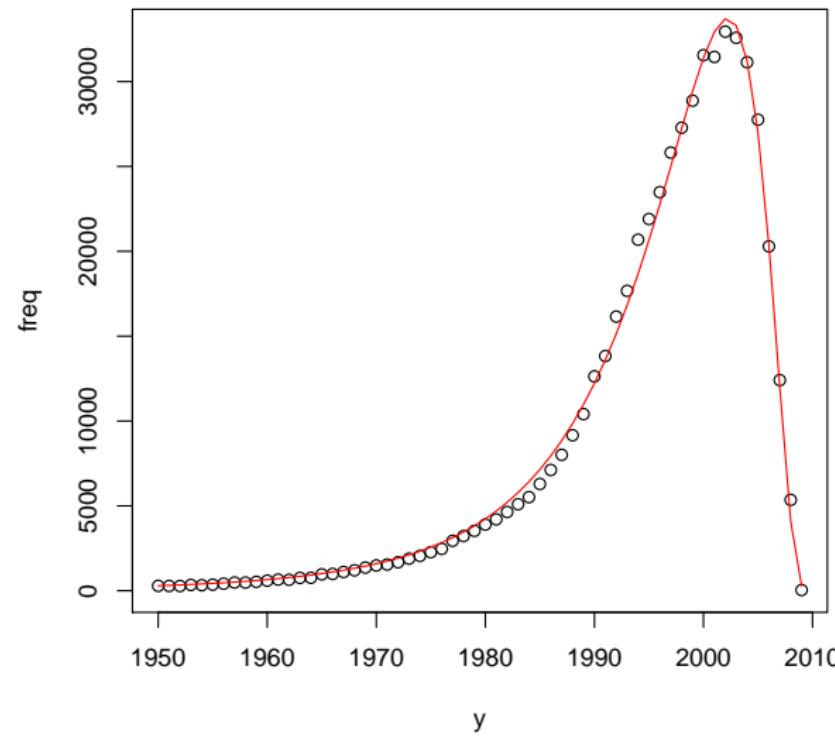
Statistics

Morphisms

Partitions

Subgraphs

Cuts





Homomorphisms of graphs

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

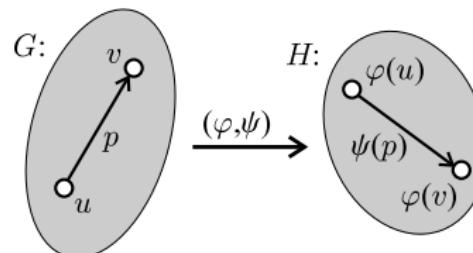
Cuts

Functions (φ, ψ) , $\varphi: \mathcal{V} \rightarrow \mathcal{V}'$ and $\psi: \mathcal{L} \rightarrow \mathcal{L}'$ determine a *weak homomorphism* of graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ in graph $\mathcal{H} = (\mathcal{V}', \mathcal{L}')$ iff:

$$\forall u, v \in \mathcal{V} \forall p \in \mathcal{L} : (p(u : v) \Rightarrow \psi(p)(\varphi(u) : \varphi(v)))$$

and they determine a *(strong) homomorphism* of graph \mathcal{G} in graph \mathcal{H} iff:

$$\forall u, v \in \mathcal{V} \forall p \in \mathcal{L} : (p(u, v) \Rightarrow \psi(p)(\varphi(u), \varphi(v)))$$



If φ and ψ are bijections and the condition hold in both direction we get an *isomorphism* of graphs \mathcal{G} and \mathcal{H} . We denote the weak isomorphism by $\mathcal{G} \sim \mathcal{H}$; and the (strong) isomorphism by $\mathcal{G} \approx \mathcal{H}$. It holds $\approx \subset \sim$.

An *invariant* of graph is called each graph characteristic that has the same value for all isomorphic graphs.

EulerGT



Homomorphism

Subnetworks

V. Batagelj

Size of networks

Pajek

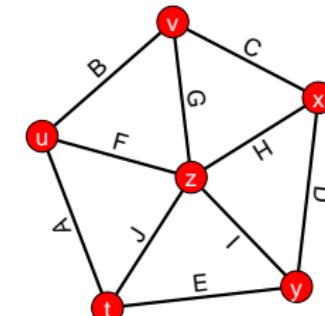
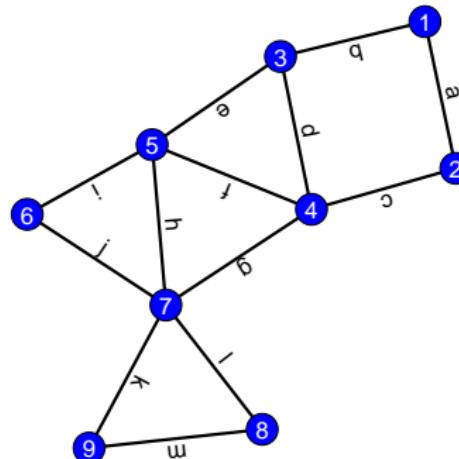
Statistics

Morphisms

Partitions

Subgraphs

Cuts



φ	1	2	3	4	5	6	7	8	9
	t	y	z	x	v	u	z	y	t

ψ	a	b	c	d	e	f	g	h	i	j	k	l	m
	E	J	D	H	G	C	H	G	B	F	J	I	E

homoEna.net



Isomorphic graphs

Subnetworks

V. Batagelj

Size of networks

Pajek

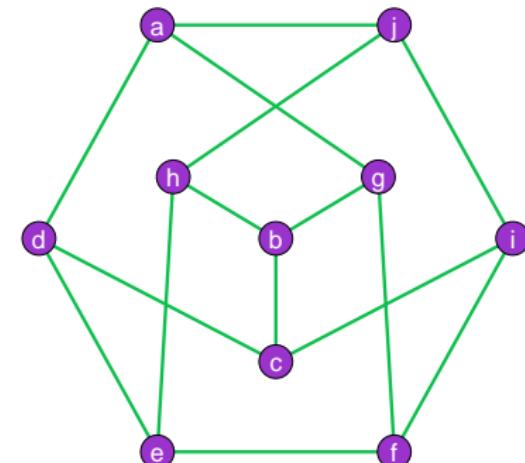
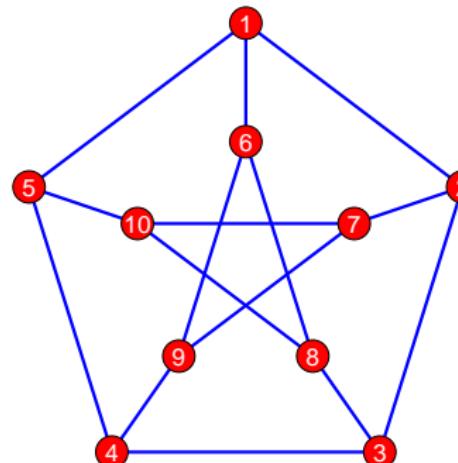
Statistics

Morphisms

Partitions

Subgraphs

Cuts



$$\varphi \begin{array}{ccccccccc} | & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline & b & h & j & a & g & c & e & i & d & f \end{array}$$

izoPet.net



Clusters, clusterings, partitions, hierarchies

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

A nonempty subset $C \subseteq \mathcal{V}$ is called a *cluster* (group). A nonempty set of clusters $\mathbf{C} = \{C_i\}$ forms a *clustering*.

Clustering $\mathbf{C} = \{C_i\}$ is a *partition* iff

$$\cup \mathbf{C} = \bigcup_i C_i = \mathcal{V} \quad \text{and} \quad i \neq j \Rightarrow C_i \cap C_j = \emptyset$$

Clustering $\mathbf{C} = \{C_i\}$ is a *hierarchy* iff

$$C_i \cap C_j \in \{\emptyset, C_i, C_j\}$$

Hierarchy $\mathbf{C} = \{C_i\}$ is *complete*, iff $\cup \mathbf{C} = \mathcal{V}$; and is *basic* if for all $v \in \cup \mathbf{C}$ also $\{v\} \in \mathbf{C}$.



Examples

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

Node set:

$$\mathcal{V} = \{a, b, c, d, e, f, g\}$$

Partition:

$$\mathbf{C} = \{\{a, b, e\}, \{c, g\}, \{d, f\}\}$$

Cluster, class:

$$C_2 = \{c, g\}$$

Hierarchy:

$$\begin{aligned}\mathbf{H} = & \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}, \\& \{a, e\}, \{c, g\}, \{d, f\}, \{a, b, e\}, \\& \{c, d, f, g\}, \{a, b, c, d, e, f, g\}\}\end{aligned}$$



Draw / Partition

Subnetworks

V. Batagelj

Size of networks

Pajek

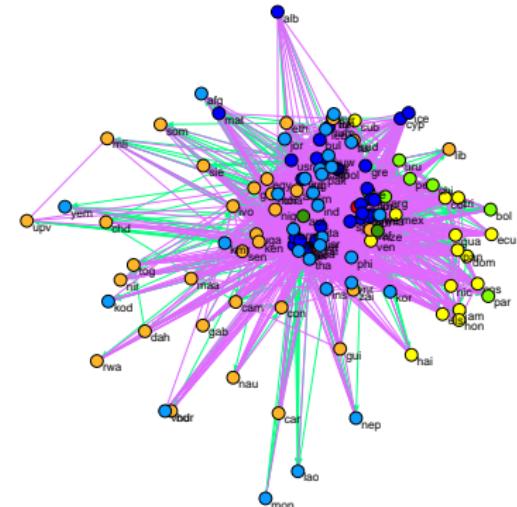
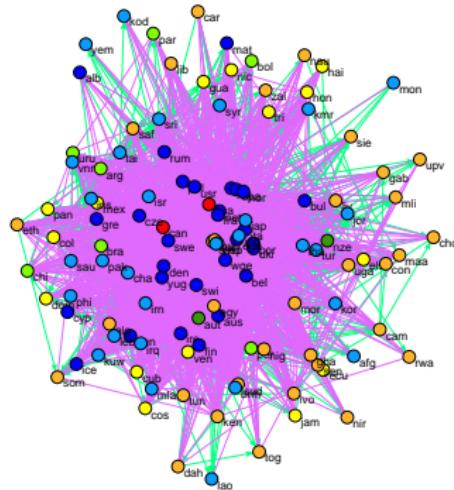
Statistics

Morphisms

Partitions

Subgraphs

Cuts



Draw/Network + First Partition
Layout/Energy/Kamada-Kawai/Free
Layout/Energy/Fruchterman Reingold/2D



Contraction of cluster

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

Contraction of cluster C is called a graph \mathcal{G}/C , in which all nodes of the cluster C are replaced by a single node, say c . More precisely:

$\mathcal{G}/C = (\mathcal{V}', \mathcal{L}')$, where $\mathcal{V}' = (\mathcal{V} \setminus C) \cup \{c\}$ and \mathcal{L}' consists of links from \mathcal{L} that have both endnodes in $\mathcal{V} \setminus C$. Beside these it contains also a 'star' with the center c and: arc (v, c) , if

$\exists p \in \mathcal{L}, u \in C : p(v, u)$; or arc (c, v) , if $\exists p \in \mathcal{L}, u \in C : p(u, v)$.

There is a loop (c, c) in c if $\exists p \in \mathcal{L}, u, v \in C : p(u, v)$.

In a network over graph \mathcal{G} we have also to specify how are determined the values/weights in the shrunk part of the network. Usually as the sum or maksimum/minimum of the original values.

Operations/Network + Partition/Shrink Network



Contracted clusters – international trade

Subnetworks

V. Batagelj

Size of networks

Pajek

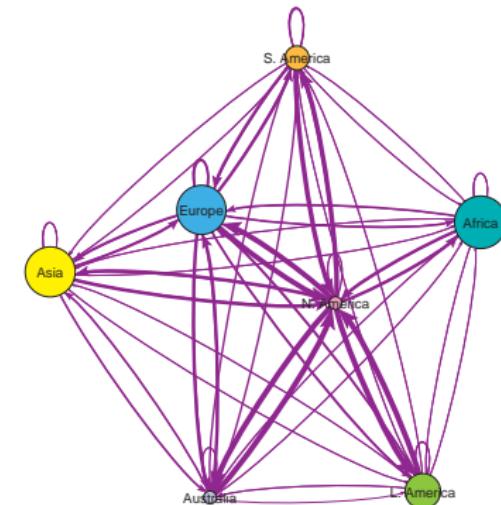
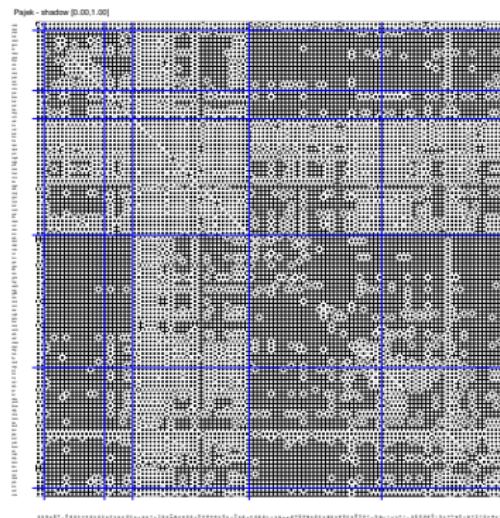
Statistics

Morphisms

Partitions

Subgraphs

Cuts



Snyder and Kick's international trade. Matrix display of dense networks.

$$w(C_i, C_j) = \frac{n(C_i, C_j)}{n(C_i) \cdot n(C_j)}$$

Macros.



Computing the weights w

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

File/Pajek Project File/Read [SKtrade.paj]
Network/Create New Network/Transform/Remove/Loops [No]
Network/Create New Network/Transform/Edges -> Arcs [No]
Operations/Network+Partition/Shrink Network [1 0]

	1	2	3	4	5	6	7	Label
	1.	2	30	13	56	42	45	#usa
	2.	30	74	25	196	20	37	#cub
	3.	12	28	33	124	16	36	#per
	4.	55	217	130	694	427	483	#uki
	5.	42	8	14	406	122	117	#mli
	6.	43	37	43	444	142	307	#irn
	7.	4	4	5	39	9	30	#aut

Partition/Make Permutation
[select partition (Sub)continents]
Operations/Partition+Permutation/
Functional Composition Partition*Permutation
Partition/Count

count 2 15 7 29 33 30 2



... Computing the weights w

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

```
Partition/Copy to Vector  
Vector/Create Constant Vector [7 1.0]  
[select as second vector Copy of partition ...]  
Vectors/Divide (First/Second)  
Network/Create Vector/Get Loops  
Vectors/Add (First+Second)  
Operations/Network+Vector/Transform/Put Loops/as Arcs  
[select vector Divide V? by ...]  
Operations/Network+Vector/Vector#Network/input  
Operations/Network+Vector/Vector#Network/output
```

	1	2	3	4	5	6	7	
#usa	1.	0.50	1.00	0.93	0.97	0.64	0.75	1.00
#cub	2.	1.00	0.33	0.24	0.45	0.04	0.08	0.40
#per	3.	0.86	0.27	0.67	0.61	0.07	0.17	0.36
#uki	4.	0.95	0.50	0.64	0.83	0.45	0.56	0.71
#mli	5.	0.64	0.02	0.06	0.42	0.11	0.12	0.17
#irn	6.	0.72	0.08	0.20	0.51	0.14	0.34	0.50
#aut	7.	1.00	0.13	0.36	0.67	0.14	0.50	0.50

Note: Set diagonal values to 1 ?

Macro `weights`.



Subgraph

Subnetworks

V. Batagelj

Size of networks

Pajek

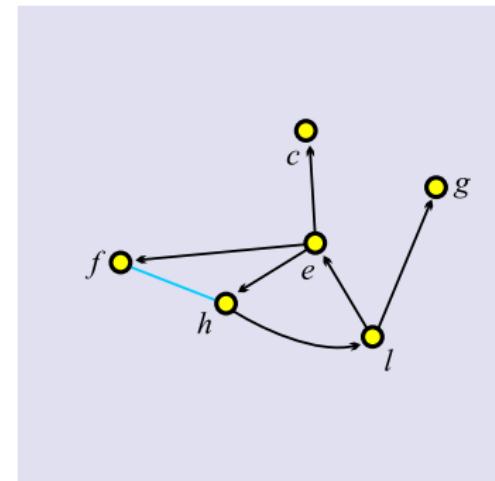
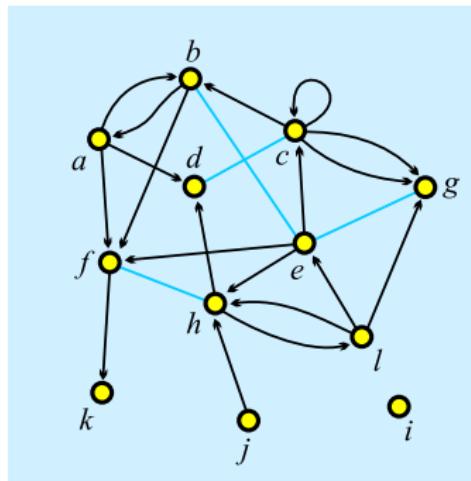
Statistics

Morphisms

Partitions

Subgraphs

Cuts



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



Subgraph in igraph

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

```
induced_subgraph(graph, vids,
                  impl=c("auto", "copy_and_delete", "create_from_scratch"))

subgraph.edges(graph, eids, delete.vertices=TRUE)

delete_edges(graph, edges)

> Class <- read.graph("class.net", format="pajek")
> vertex_attr_names(Class)
[1] "id"      "name"    "x"       "y"       "z"
> vertex_attr(Class)$shape <- NULL
> sex <- as.integer(substr(vertex_attr(Class)$id,1,1)=="m")
> F <- V(Class)[sex==0]
> Fclass <- induced_subgraph(Class,F)
> plot(Fclass)
> N <- E(Class)[F %--% F]
> N
+ 30/56 edges from 3a5cb23 (vertex names):
[1] w07->w42 w09->w24 w09->w10 w10->w28 w24->w10 w28->w42 w42->
[9] w12->w63 w09->w12 w07->w10 w07->w22 w07->w28 w10->w22 w22->
[17] w22->w28 w24->w42 w09->w63 w63->w12 w12->w09 w10->w07 w22->
[25] w22->w10 w24->w22 w42->w22 w28->w22 w42->w24 w63->w09
```



Cut-out – induced subgraph: Snyder and Kick Africa

Subnetworks

V. Batagelj

Size of networks

Pajek

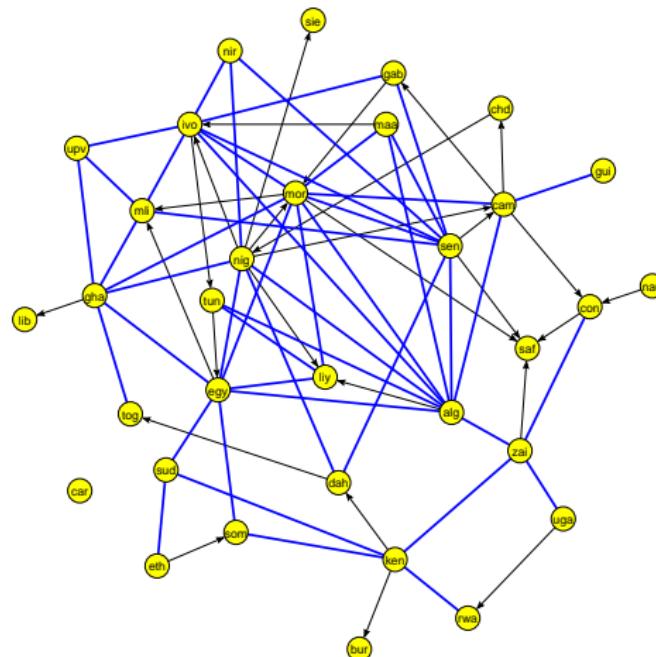
Statistics

Morphisms

Partitions

Subgraphs

Cuts



Operations/Network + Partition/Extract
Subnetwork [6]



Cut-out: Snyder and Kick

Latin America : South America

Subnetworks

V. Batagelj

Size of networks

Pajek

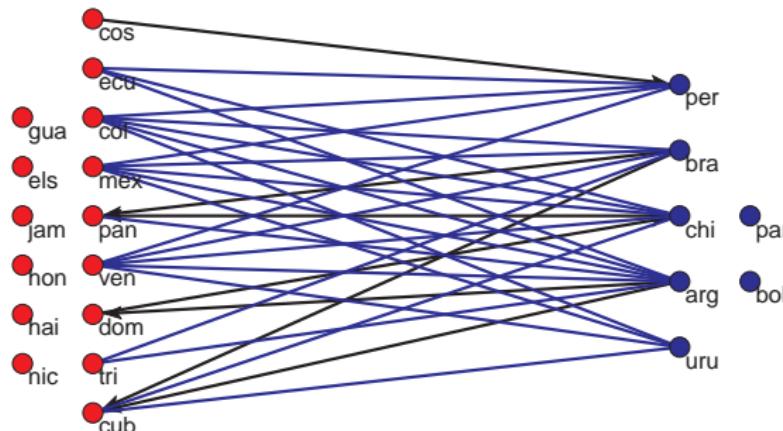
Statistics

Morphisms

Partitions

Subgraphs

Cuts



Operations/Network + Partition/Extract Subnetwork [3, 4]
Operations/Network + Partition/Transform/Remove lines/
Inside clusters [3, 4]

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

[Draw] Move/Grid



Cut-outs in igraph

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

```
extract_clusters <- function(N,atn,clus){  
  C <- vertex_attr(N,atn); S <- V(N)[C %in% clus]  
  return(induced_subgraph(N,S))  
}  
interlinks <- function(N,atn,c1,c2,col1="red",col2="blue"){  
  S <- extract_clusters(N,atn,c(c1,c2))  
  C <- vertex_attr(S,atn)  
  C1 <- V(S)[C==c1]; C2 <- V(S)[C==c2]  
  V(S)$color <- ifelse(C==c1,col1,col2)  
  P <- E(S)[(C1 %--% C1) | (C2 %--% C2)]  
  return(delete_edges(S,P))  
}  
  
> library(igraph); source("igraph+.R")  
> SaK <- read.graph("./nets/SaKtrade.net", format="pajek")  
> V(SaK)$sc <- read_Pajek_clu("./nets/SaKtrade.clu", skip=7)  
> Af <- extract_clusters(SaK, "sc", c(6))  
> plot(Af)  
> B <- interlinks(SaK, "sc", 3, 4, col1="yellow", col2="cyan")  
> plot(B)
```



Cuts

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

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 $\mathcal{N} = (\mathcal{V}, \mathcal{L}, p)$, $p : \mathcal{V} \rightarrow \mathbb{R}$, at selected level t is a subnetwork $\mathcal{N}(t) = (\mathcal{V}', \mathcal{L}(\mathcal{V}'), p)$, 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 $\mathcal{N} = (\mathcal{V}, \mathcal{L}, w)$, $w : \mathcal{L} \rightarrow \mathbb{R}$, at selected level t is a subnetwork $\mathcal{N}(t) = (\mathcal{V}(\mathcal{L}'), \mathcal{L}', w)$, 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}' .



Node-cut: Krebs Internet Industries, core=6

Subnetworks

V. Batagelj

Size of networks

Pajek

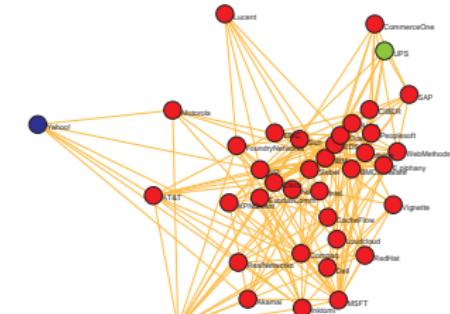
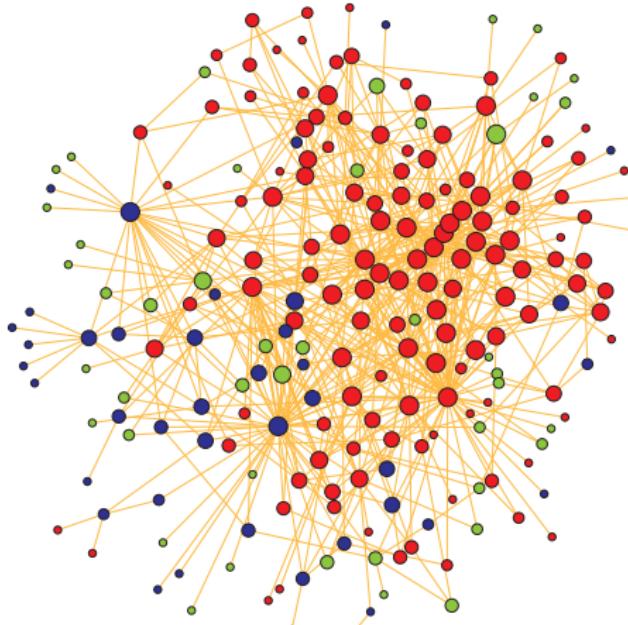
Statistics

Morphisms

Partitions

Subgraphs

Cuts



Each node represents a company that competes in the Internet industry, 1998 do 2001. $n = 219$, $m = 631$.
red – content, blue – infrastructure, green – commerce. Two companies are linked with an edge if they have announced a joint venture, strategic alliance or other partnership.



Triangular network

Subnetworks

V. Batagelj

Size of networks

Pajek

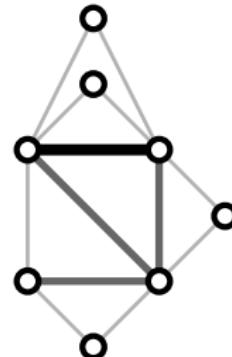
Statistics

Morphisms

Partitions

Subgraphs

Cuts



Let \mathcal{G} be a simple undirected graph. A *triangular network* $\mathcal{N}_T(\mathcal{G}) = (\mathcal{V}, \mathcal{E}_T, w)$ determined by \mathcal{G} is a subgraph $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}_T)$ of \mathcal{G} whose set of edges \mathcal{E}_T consists of all triangular edges of $\mathcal{E}(\mathcal{G})$. For $e \in \mathcal{E}_T$ the weight $w(e)$ equals to the number of different triangles in \mathcal{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 \mathcal{G} then $w(e) \geq k - 2$.

Network/Create New Network/with Ring Counts/3-Rings



Link-cut: Krebs Internet Industries, $w_3 \geq 5$

Subnetworks

V. Batagelj

Size of networks

Pajek

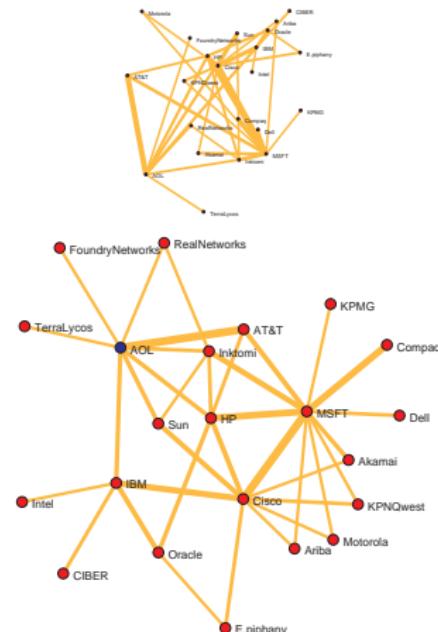
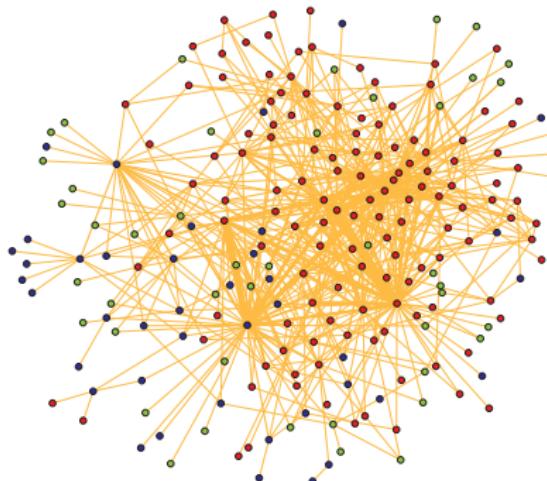
Statistics

Morphisms

Partitions

Subgraphs

Cuts





Overlap weight – definition

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

The (topological) *overlap weight* of an edge $e = (u : v) \in \mathcal{E}$ in an undirected simple graph $\mathbf{G} = (\mathcal{V}, \mathcal{E})$ is defined as

$$o(e) = \frac{t(e)}{(\deg(u) - 1) + (\deg(v) - 1) - t(e)}$$

$t(e) = w_3(e)$ is the *number of triangles* (cycles of length 3) to which the edge e belongs. In the case $\deg(u) = \deg(v) = 1$ we set $o(e) = 0$.

The overlap weight is essentially a Jaccard similarity index

$$J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

for $X = N(u) \setminus \{v\}$ and $Y = N(v) \setminus \{u\}$ where $N(z)$ is the set of neighbors of a node z .

Denoting $\mu = \max_{e \in \mathcal{E}} t(e)$ and $M(e) = \max(\deg(u), \deg(v)) - 1$ we define a *corrected overlap weight* as

$$o'(e) = \frac{t(e)}{\mu + M(e) - t(e)}$$



Cuts in Pajek

Subnetworks

V. Batagelj

Size of networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

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 [+10] [#10]
Vector/Make Partition/by Intervals/Selected Thresholds [t]
Operations/Network + Partition/Extract Subnetwork [2]
```

Link-cut: weighted network

```
Network/Info/Line values [#10]
Network/Create New Network/Transform/Remove/Lines with Value/
    lower than [t]
Network/Create Partition/Degree/All
Operations/Network + Partition/Extract Subnetwork [1-*]
```



Cuts in igraph

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

```
vertex_cut <- function(N,atn,t) {
  v <- vertex_attr(N,atn); vCut <- V(N) [v>=t]
  return(induced_subgraph(N,vCut))
}
edge_cut <- function(N,atn,t) {
  w <- edge_attr(N,atn); eCut <- E(N) [w>=t]
  return(subgraph.edges(N,eCut))
}

> R <- read.graph("./nets/class.net",format="pajek")
> vertex_attr(R)$shape <- NULL
> V(R)$deg <- degree(R)
> Cut <- vertex_cut(R,"deg",8)
> plot(Cut,vertex.size=V(Cut)$deg*3)
> E(R)$rnd <- sample(1:10,ecount(R),replace=TRUE)
> Ec <- edge_cut(R,"rnd",9)
> plot(Ec,edge.width=E(Ec)$rnd)
```



Simple analysis using cuts

Subnetworks

V. Batagelj

Size of
networks

Pajek

Statistics

Morphisms

Partitions

Subgraphs

Cuts

We look at the components of $\mathcal{N}(t)$. 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 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

Subnetworks

V. Batagelj

Size of networks

Pajek

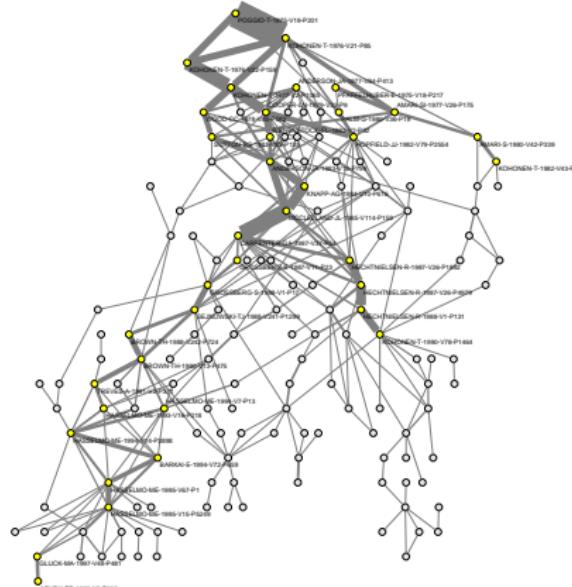
Statistics

Morphisms

Partitions

Subgraphs

Cuts



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 nodes, 12731 arcs).

See [paper](#).



Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

Introduction to Network Analysis using **Pajek**

4. Structure of networks: connectivity

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

PhD and MS program in Statistics
University of Ljubljana, 2022



Outline

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering

- 1 Connectivity
- 2 Condensation
- 3 Bow-tie
- 4 Other connectivities
- 5 Important nodes
- 6 Closeness
- 7 Betweenness
- 8 Hubs and authorities
- 9 Clustering



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

Current version of slides (March 23, 2022 at 00 :11): [slides PDF](#)



Walks

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

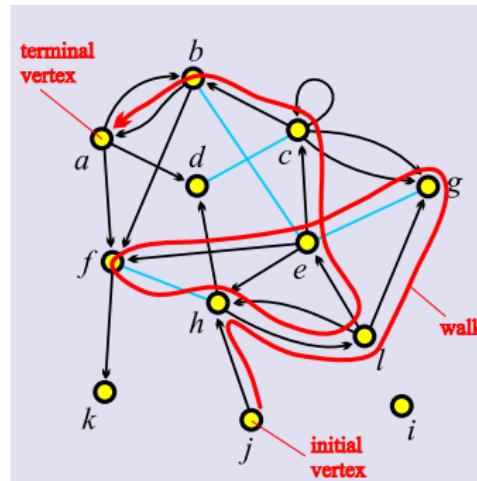
Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering



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.



Shortest paths

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

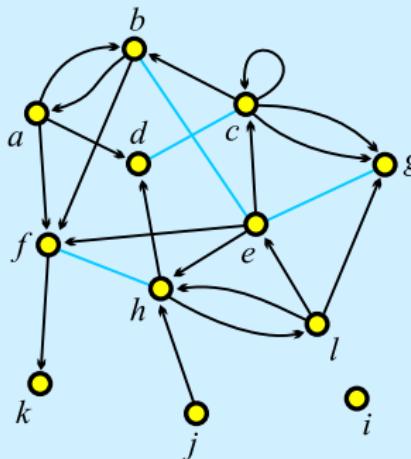
Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering



A shortest path from u to v is also called a *geodesic* from u to v . Its length is denoted by $d(u, v)$.

If there is no walk from u to v then $d(u, v) = \infty$.

$$d(j, a) = |(j, h, d, c, b, a)| = 5$$

$$d(a, j) = \infty$$

$\hat{d}(u, v) = \max(d(u, v), d(v, u))$ is a *distance*:

$$\hat{d}(v, v) = 0, \hat{d}(u, v) = \hat{d}(v, u),$$

$$\hat{d}(u, v) \leq \hat{d}(u, t) + \hat{d}(t, v).$$

The *diameter* of a graph equals to the distance between the most distant pair of nodes: $D = \max_{u, v \in V} d(u, v)$.

Network/Create New Network/Subnetwork with
Paths/



Shortest paths

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

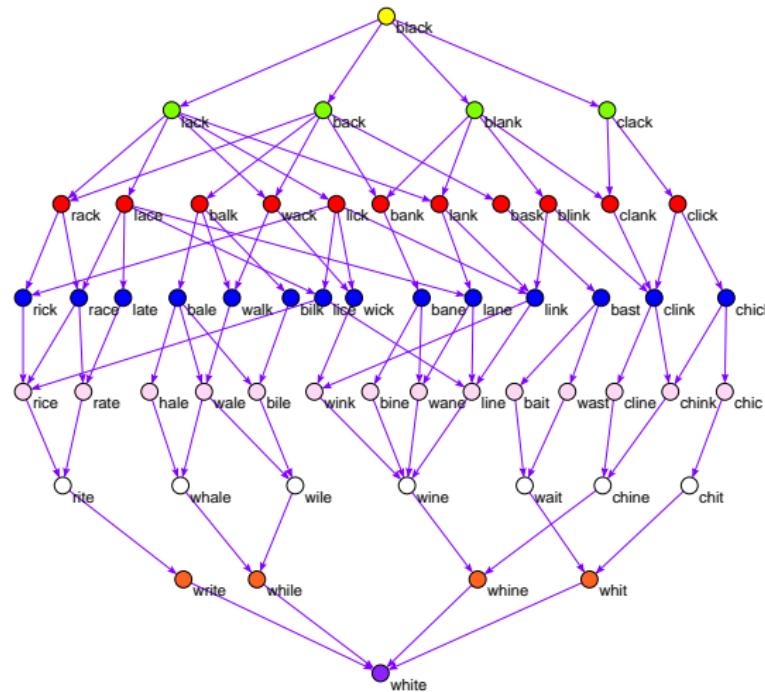
Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering



DICT28.



Equivalence relations and Partitions

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering

A relation R on \mathcal{V} is an *equivalence* relation iff it is reflexive $\forall v \in \mathcal{V} : vRv$, symmetric $\forall u, v \in \mathcal{V} : uRv \Rightarrow vRu$, and transitive $\forall u, v, z \in \mathcal{V} : uRz \wedge zRv \Rightarrow uRv$.

Each equivalence relation determines a partition into *equivalence classes* $[v] = \{u : vRu\}$.

Each partition \mathbf{C} determines an equivalence relation $uRv \Leftrightarrow \exists C \in \mathbf{C} : u \in C \wedge v \in C$.

k-neighbors of v is the set of nodes on 'distance' k from v , $N^k(v) = \{u \in \mathcal{V} : d(v, u) = k\}$.

The set of all k -neighbors, $k = 0, 1, \dots$ of v is a partition of \mathcal{V} .

k-neighborhood of v , $N^{(k)}(v) = \{u \in \mathcal{V} : d(v, u) \leq k\}$.

Network/Create Partition/k-Neighbors



Motorola's neighborhood

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

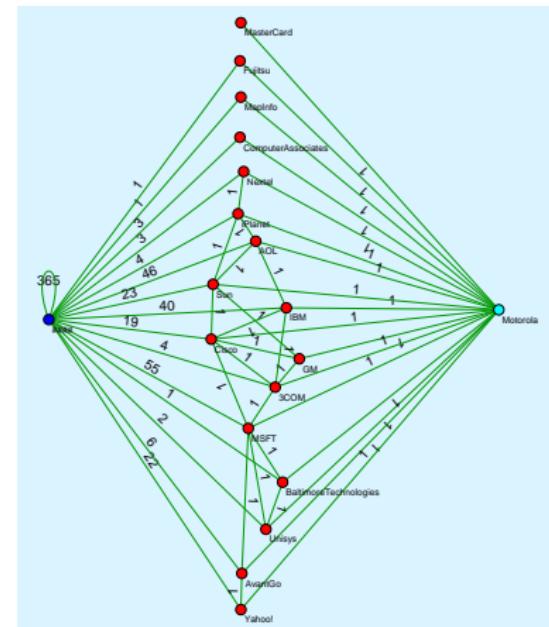
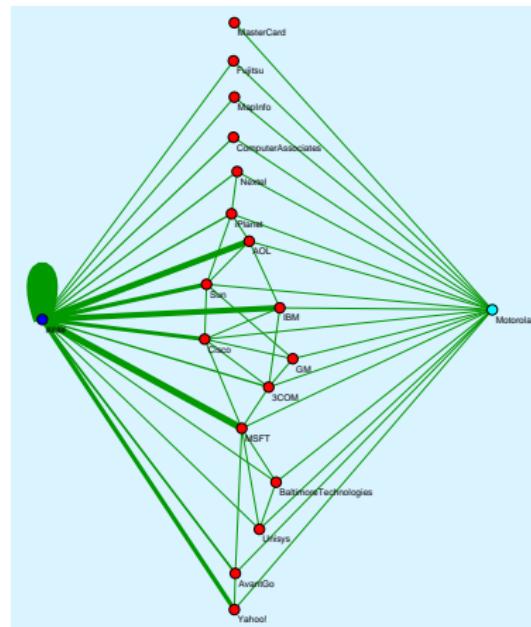
Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering



The thickness of edges is a square root of its value.



Connectivity

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

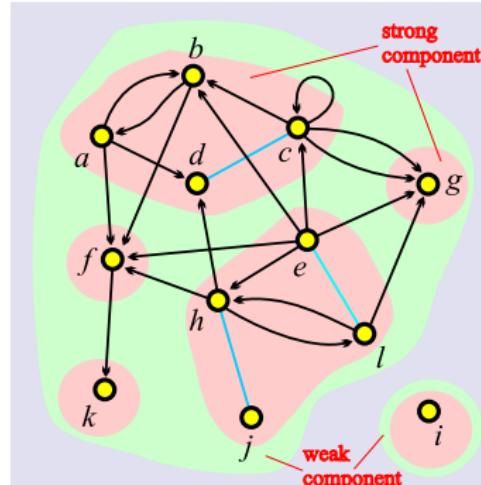
Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering



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/



Connectivity in igraph

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

```
> wdir <- "C:/..../2017/Moscow/Rnet/test"  
> setwd(wdir)  
> library(igraph)  
> source("C:\\\\...\\\\Rnet\\\\test\\\\igraph+.R")  
> R <- read.graph("./nets/class.net", format="pajek")  
> vertex_attr(R)$shape <- NULL  
> plot(R)  
> w <- components(R, mode="weak")  
> w  
> s <- components(R, mode="strong")  
> s  
> V(R)$strong <- s$membership  
> col <- c("red", "green", "orange", "blue", "green", "magenta",  
"grey", "black")  
> plot(R, vertex.color=col[s$membership])  
> main <- extract_clusters(R, "strong", c(4))  
> plot(main)
```



Weak components

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

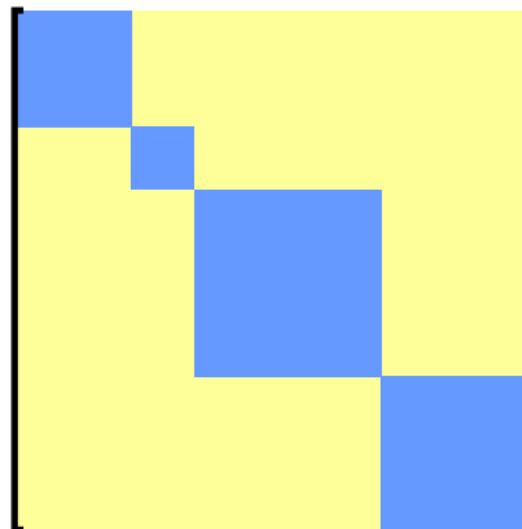
Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering



Reordering the nodes of network such that the nodes from the same class of weak partition are put together we get a matrix representation consisting of diagonal blocks – weak components.

Most problems can be solved separately on each component and afterward these solutions combined into final solution.



Special graphs – bipartite, tree

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

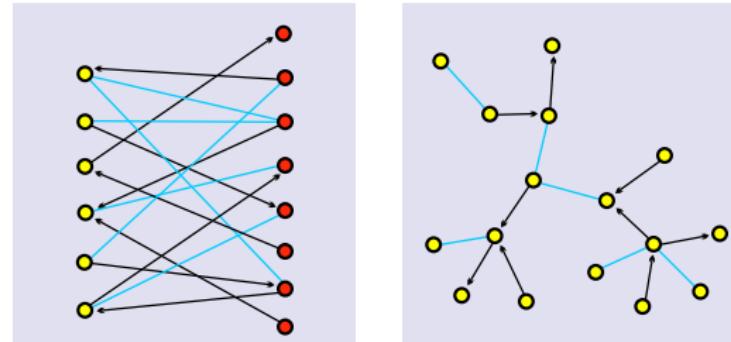
Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering



A graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ is *bipartite* iff its set of nodes \mathcal{V} can be partitioned into two sets \mathcal{V}_1 and \mathcal{V}_2 such that every link from \mathcal{L} has one end-node in \mathcal{V}_1 and the other in \mathcal{V}_2 .

A weakly connected graph \mathcal{G} is a *tree* iff it doesn't contain loops and semicycles of length at least 3.



Condensation

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

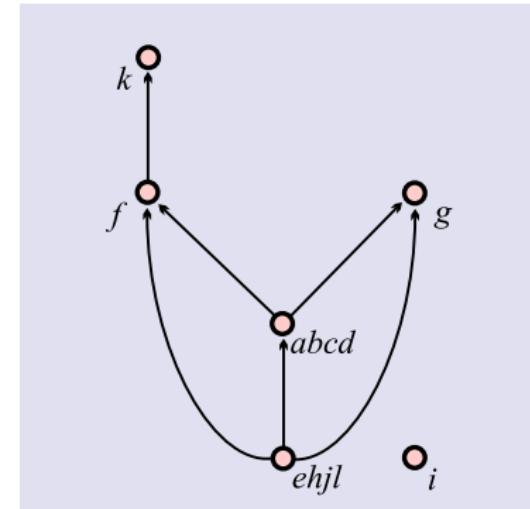
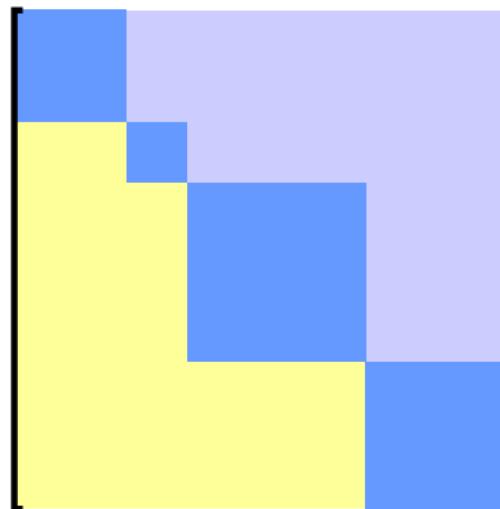
Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering



If we shrink every strong component of a given graph into a node, delete all loops and identify parallel arcs the obtained *reduced* graph is acyclic. For every acyclic graph an *ordering / level* function $i : \mathcal{V} \rightarrow \mathbb{N}$ exists s.t. $(u, v) \in \mathcal{A} \Rightarrow i(u) < i(v)$.



Condensation – Example

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

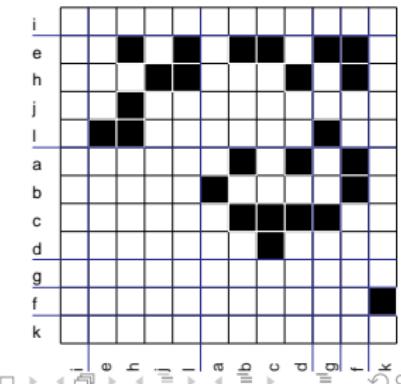
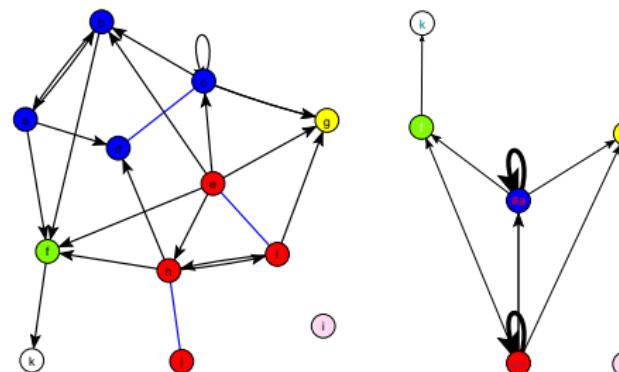
Closeness

Betweenness

Hubs and
authorities

Clustering

```
Network/Create Partition/Components/Strong [1]
Operations/Network+Partition/Shrink Network [1][0]
Network/Create New Network/Transform/Remove/Loops [yes]
Network/Acyclic Network/Depth Partition/Acyclic
Partition/Make Permutation
Permutation/Inverse Permutation
select partition [Strong Components]
Operations/Partition+Permutation/Functional Composition Partition
Partition/Make Permutation
select [original network]
File/Network/Export Matrix to EPS/Using Permutation
```





Internal structure of strong components

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

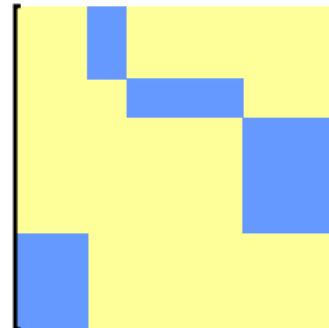
Important nodes

Closeness

Betweenness

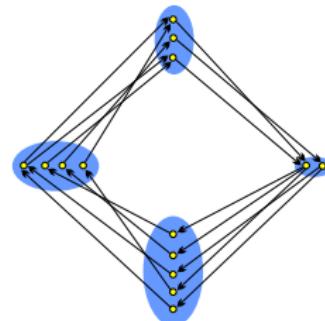
Hubs and authorities

Clustering



Let d be the largest common divisor of lengths of closed walks in a strong component.

The component is said to be *simple*, iff $d = 1$; otherwise it is *periodical* with a period d .



The set of nodes \mathcal{V} of strongly connected directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{R})$ can be partitioned into d clusters $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_d$, s.t. for every arc $(u, v) \in \mathcal{R}$ holds $u \in \mathcal{V}_i \Rightarrow v \in \mathcal{V}_{(i \bmod d) + 1}$.

Network/Create Partition/
Components/Strong-Periodic



... Internal structure of strong components

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

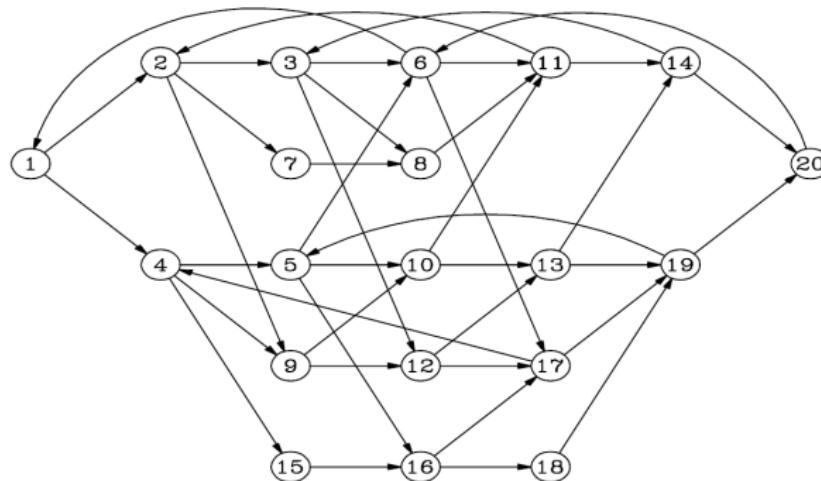
Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering



Bonhoure's periodical graph. Pajek data



Bow-tie structure of the Web graph

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

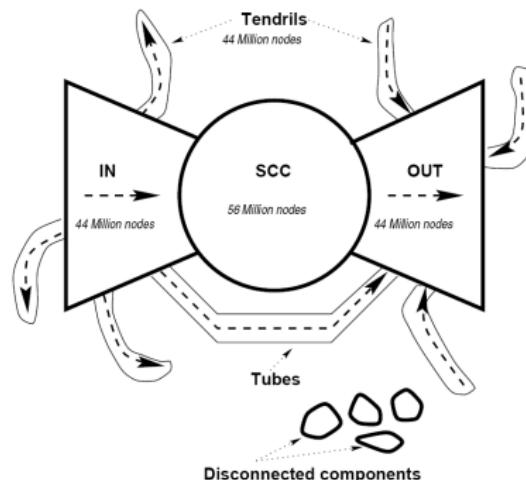
Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering



Kumar &: The Web as a graph

Let \mathcal{S} be the *largest strong component* in network \mathcal{N} ; \mathcal{W} the weak component containing \mathcal{S} ; \mathcal{I} the set of nodes from which \mathcal{S} can be reached; \mathcal{O} the set of nodes reachable from \mathcal{S} ; \mathcal{T} (tubes) set of nodes (not in \mathcal{S}) on paths from \mathcal{I} to \mathcal{O} ; $\mathcal{R} = \mathcal{W} \setminus (\mathcal{I} \cup \mathcal{S} \cup \mathcal{O} \cup \mathcal{T})$ (tendrils); and $\mathcal{D} = \mathcal{V} \setminus \mathcal{W}$. The partition

$$\{\mathcal{I}, \mathcal{S}, \mathcal{O}, \mathcal{T}, \mathcal{R}, \mathcal{D}\}$$

is called the *bow-tie* partition of \mathcal{V} .

Note: chains can exist in the set \mathcal{R} .

Network/Create Partition/Bow-Tie



Biconnectivity

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

Nodes u and v are *biconnected* iff they are connected (in both directions) by two independent (no common internal node) paths. Biconnectivity determines a partition of the set of links.

A node is an *articulation* node iff its deletion increases the number of weak components in a graph.

A link is a *bridge* iff its deletion increases the number of weak components in a graph.

Network/Create New Network/with Bi-Connected Components.

The notion of biconnectivity can be generalized do k -connectivity. No very efficient algorithm for $k > 3$ exists.



k -connectivity

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

Node connectivity κ of graph \mathcal{G} is equal to the smallest number of nodes that, if deleted, induce a disconnected graph or the trivial graph K_1 .

Link connectivity λ of graph \mathcal{G} is equal to the smallest number of links that, if deleted, induce a disconnected graph or the trivial graph K_1 .

Whitney's inequality: $\kappa(\mathcal{G}) \leq \lambda(\mathcal{G}) \leq \delta(\mathcal{G})$.

Graph \mathcal{G} is **(node) k -connected**, if $\kappa(\mathcal{G}) \geq k$ and is **link k -connected**, if $\lambda(\mathcal{G}) \geq k$.

Whitney / Menger theorem: Graph \mathcal{G} is node/link k -connected iff every pair of nodes can be connected with k node/link internally disjoint (semi)walks.



Triangular and short cycle connectivities

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

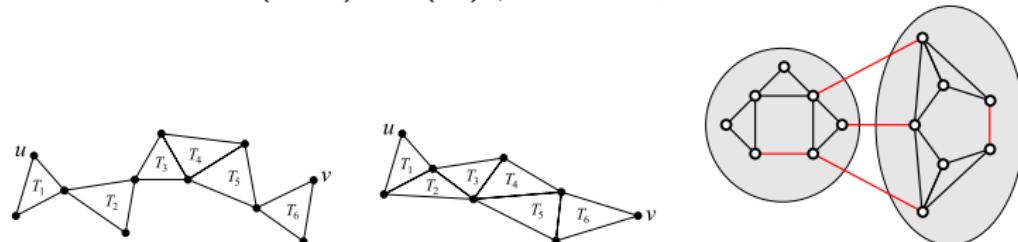
Betweenness

Hubs and
authorities

Clustering

In an undirected graph we call a *triangle* a subgraph isomorphic to K_3 .

A sequence (T_1, T_2, \dots, T_s) of triangles of \mathcal{G} (*node*) *triangularly connects* nodes $u, v \in \mathcal{V}$ iff $u \in T_1$ and $v \in T_s$ or $u \in T_s$ and $v \in T_1$ and $\mathcal{V}(T_{i-1}) \cap \mathcal{V}(T_i) \neq \emptyset$, $i = 2, \dots, s$. It *edge triangularly connects* nodes $u, v \in \mathcal{V}$ iff a stronger version of the second condition holds $\mathcal{E}(T_{i-1}) \cap \mathcal{E}(T_i) \neq \emptyset$, $i = 2, \dots, s$.



Node triangular connectivity is an equivalence on \mathcal{V} ; and edge triangular connectivity is an equivalence on \mathcal{E} . See the [paper](#).



Triangular connectivity in directed graphs

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

Important nodes

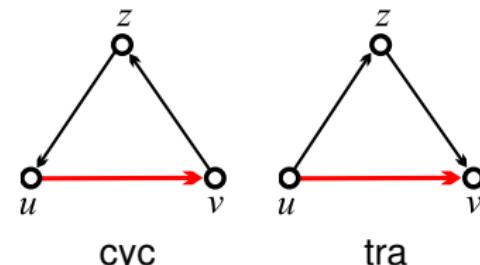
Closeness

Betweenness

Hubs and authorities

Clustering

If a graph \mathcal{G} is mixed we replace edges with pairs of opposite arcs. In the following let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ 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: **cyclic** and **transitive**.



For each type we get the corresponding triangular network \mathcal{N}_{cyc} and \mathcal{N}_{tra} by determining the corresponding weight w_{cyc} or w_{tra} to its arcs, counting the number of cyclic/transitive triangles that contain the arc. We remove arcs with weight zero. The notion of triangular connectivity can be extended to the notion of **short (semi) cycle connectivity**.

Network/Create New Network/with Ring Counts/3-Rings/Directed



Edge-cut at level 16 of triangular network of Erdős collaboration graph

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

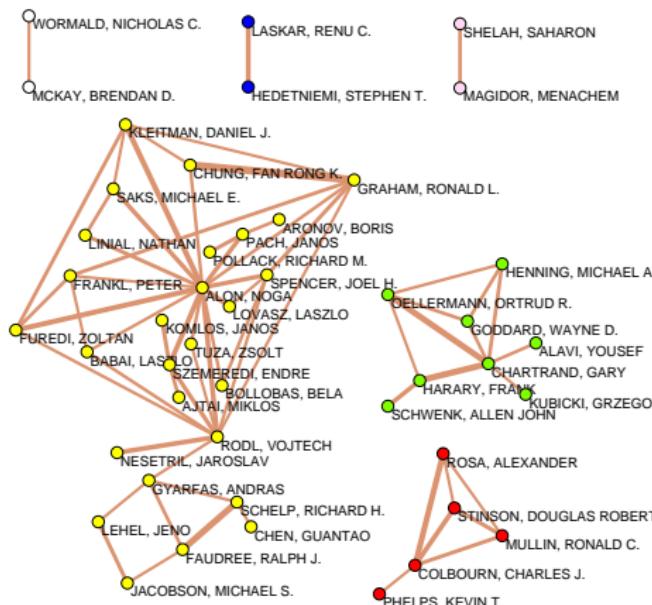
Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering



without Erdős,
 $n = 6926$,
 $m = 11343$



Arc-cut at level 11 of transitive triangular network of ODLIS dictionary

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

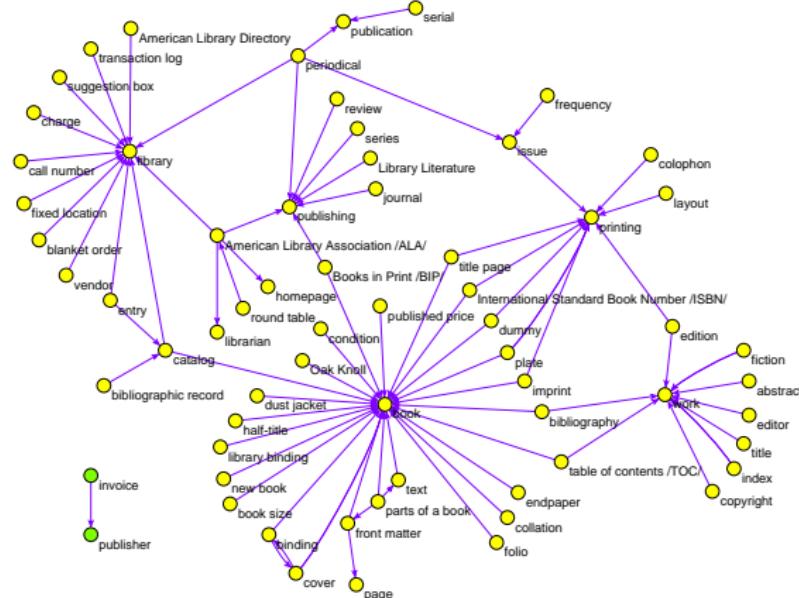
Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering





Important nodes in network

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

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

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

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

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



... Important nodes in network

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

This gives us two main types of indices:

- networks containing directed links (we replace edges by pairs of opposite arcs): measures of *importance*; with two subgroups: measures of *influence*, based on out-going arcs; and measures of *support*, based on incoming arcs;
- measures of *centrality*, based on all links.

For undirected networks all three types of measures coincide. If we change the direction of all arcs (replace the relation with its inverse relation) the measure of influence becomes a measure of support, and vice versa.



... Important nodes in network

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

The real meaning of measure of importance depends on the relation described by a network. For example the most 'important' person for the relation '... doesn't like to work with ...' is in fact the least popular person.

Removal of an important node/link from a network produces a substantial change in structure/functioning of the network.



Normalization

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

Let $p : \mathcal{V} \rightarrow \mathbb{R}$ be an index in network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, p)$. If we want to compare indices p over different networks we have to make them comparable. Usually we try to achieve this by *normalization* of p . Let $\mathcal{N} \in \mathbf{N}(\mathcal{V})$, where $\mathbf{N}(\mathcal{V})$ is a selected family of networks over the same set of nodes \mathcal{V} ,

$$p_{\max} = \max_{\mathcal{N} \in \mathbf{N}(\mathcal{V})} \max_{v \in \mathcal{V}} p_{\mathcal{N}}(v) \quad \text{and} \quad p_{\min} = \min_{\mathcal{N} \in \mathbf{N}(\mathcal{V})} \min_{v \in \mathcal{V}} p_{\mathcal{N}}(v)$$

then we define the normalized index as

$$p'(v) = \frac{p(v) - p_{\min}}{p_{\max} - p_{\min}} \in [0, 1]$$



Degrees

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

The simplest index are the degrees of nodes. Since for simple networks $\deg_{min} = 0$ and $\deg_{max} = n - 1$, the corresponding normalized indices are

$$\text{centrality} \quad \deg'(v) = \frac{\deg(v)}{n - 1}$$

and similarly

$$\text{support} \quad \text{indeg}'(v) = \frac{\text{indeg}(v)}{n}$$

$$\text{influence} \quad \text{outdeg}'(v) = \frac{\text{outdeg}(v)}{n}$$

Instead of degrees in original network we can consider also the degrees with respect to the reachability relation (transitive closure).

Network/Create Partition/Degree

Network/Create Vector/Centrality/Degree

Network/Create Vector/Centrality/Proximity

Prestige



Closeness

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

Most indices are based on the distance $d(u, v)$ between nodes in a network $\mathcal{N} = (\mathcal{V}, \mathcal{L})$. Two such indices are

$$\text{radius} \quad r(v) = \max_{u \in \mathcal{V}} d(v, u)$$

$$\text{total closeness} \quad S(v) = \sum_{u \in \mathcal{V}} d(v, u)$$

These two indices are measures of influence – to get measures of support we have to replace in definitions $d(u, v)$ with $d(v, u)$.

If the network is not strongly connected r_{\max} and S_{\max} are equal to ∞ . Sabidussi (1966) introduced a related measure $1/S(v)$; or in its normalized form

$$\text{closeness} \quad c(v) = \frac{n - 1}{\sum_{u \in \mathcal{V}} d(v, u)}$$

$D = \max_{u, v \in \mathcal{V}} d(v, u)$ is called the **diameter** of network.

Network/Create Vector/Centrality/Closeness

Network/Create New Network/Subnetwork with
Paths/Info on Diameter



Betweenness

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering

Important are also the nodes that can control the information flow in the network. If we assume that this flow uses only the shortest paths (geodesics) we get a measure of *betweenness* (Anthonisse 1971, Freeman 1977)

$$b(v) = \frac{1}{(n-1)(n-2)} \sum_{\substack{u,t \in V : g_{u,t} > 0 \\ u \neq v, t \neq v, u \neq t}} \frac{g_{u,t}(v)}{g_{u,t}}$$

where $g_{u,t}$ is the number of geodesics from u to t ; and $g_{u,t}(v)$ is the number of those among them that pass through node v .

For computation of geodesic matrix see **Brandes**.

Network/Create Vector/Centrality/Betweenness



Padgett's Florentine families

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

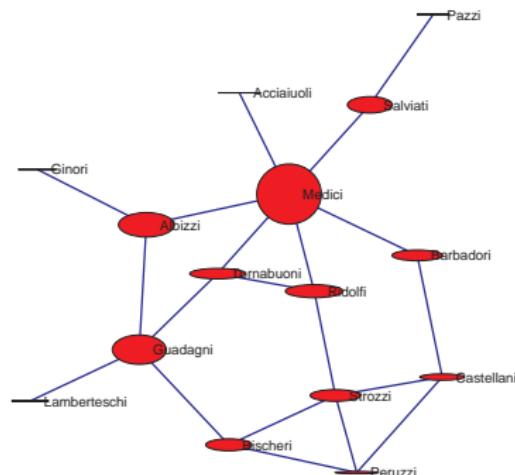
Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering



	close	between
1. Acciaiuoli	0.368421	0.000000
2. Albizzi	0.482759	0.212454
3. Barbadori	0.437500	0.093407
4. Bischeri	0.400000	0.104396
5. Castellani	0.388889	0.054945
6. Ginori	0.333333	0.000000
7. Guadagni	0.466667	0.254579
8. Lamberteschi	0.325581	0.000000
9. Medici	0.560000	0.521978
10. Pazzi	0.285714	0.000000
11. Peruzzi	0.368421	0.021978
12. Ridolfi	0.500000	0.113553
13. Salviati	0.388889	0.142857
14. Strozzi	0.437500	0.102564
15. Tornabuoni	0.482759	0.091575



Hubs and authorities

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering

To each node v of a network $\mathcal{N} = (\mathcal{V}, \mathcal{L})$ 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](#)).

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

Let \mathbf{W} be a matrix of network \mathcal{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 $\mathbf{y} = [1, 1, \dots, 1]$ 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

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

Similar procedures are used in search engines on the web to evaluate the importance of web pages.

PageRank, PageRank / Google, HITS / AltaVista, SALSA, theory.

Network/Create New Network/Subnetwork with Paths/Info on
Network/Create Vector/Centrality/Closeness
Network/Create Vector/Centrality/Betweenness
Network/Create Vector/Centrality/Hubs-Authorities
Network/Create Vector/Centrality/Clustering Coefficients

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

There exist other measures based on eigen-values and eigen-vectors such as **Katz**, **Bonachich** and **Brandes**. See also **Borgatti**.



... Hubs and authorities: football

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

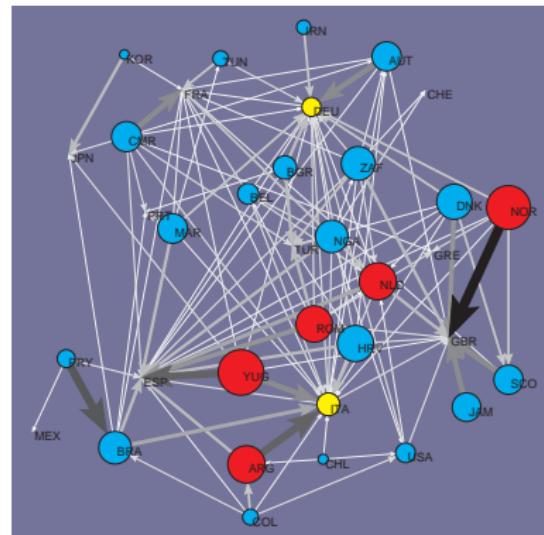
Important nodes

Closeness

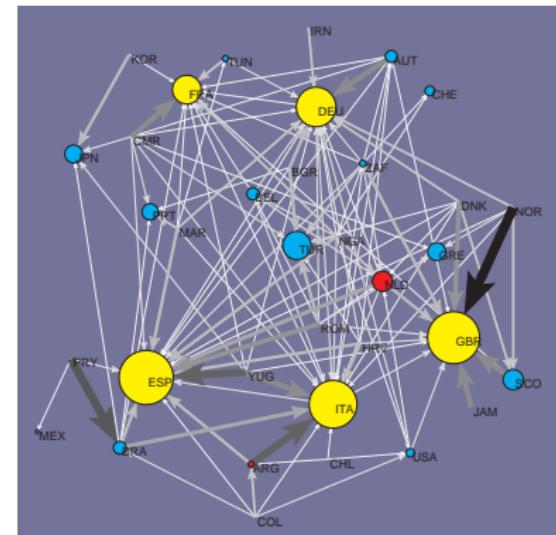
Betweenness

Hubs and authorities

Clustering



Exporters (hubs)



Importers (authorities)



Clustering

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be simple undirected graph. *Clustering* in node v is usually measured as a quotient between the number of links in subgraph $\mathcal{G}^1(v) = \mathcal{G}(N^1(v))$ induced by the neighbors of node v and the number of links in the complete graph on these nodes:

$$C(v) = \begin{cases} \frac{2|\mathcal{L}(\mathcal{G}^1(v))|}{\deg(v)(\deg(v) - 1)} & \deg(v) > 1 \\ 0 & \text{otherwise} \end{cases}$$

We can consider also the size of node neighborhood by the following correction

$$C_1(v) = \frac{\deg(v)}{\Delta} C(v)$$

where Δ is the maximum degree in graph \mathcal{G} . This measure attains its largest value in nodes that belong to an isolated clique of size Δ .



User defined indices

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other connectivities

Important nodes

Closeness

Betweenness

Hubs and authorities

Clustering

Xingqin Qi et al. defined in their paper **Terrorist Networks, Network Energy and Node Removal** a new measure of centrality based on Laplacian energy – *Laplacian centrality*

$$L(v) = \deg(v)(\deg(v) + 1) + 2 \sum_{u \in N(v)} \deg(u)$$

```
select the network
Network/Create Vector/Centrality/Degree/All
Operations/Network+Vector/Neighbours/Sum/All [False]
Vector/Transform/Multiply by [2]
select the degree vector as First
select the degree vector as Second
Vectors/Multiply (First*Second)
Vectors/Add (First+Second)
select the 2*sum on neighbors as Second
Vectors/Add (First+Second)
dispose auxiliary vectors
File/Vector/Change Label [Laplace All centrality]
```

macro



Network centralization measures

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

Extremal approach: Let $p : \mathcal{V} \rightarrow \mathbb{R}$ be an index in network $\mathcal{N} = (\mathcal{V}, \mathcal{L})$. We introduce the quantities

$$p^* = \max_{v \in \mathcal{V}} p(v)$$

$$D = \sum_{v \in \mathcal{V}} (p^* - p(v))$$

$$D^* = \max\{D(\mathcal{N}) : \mathcal{N} \text{ is a network on } \mathcal{V}\}$$

Then we can define *centralization* with respect to p

$$C_p(\mathcal{N}) = \frac{D(\mathcal{N})}{D^*}$$

Usually the most centralized graph is the star S_n and the least centralized is the complete graph K_n .



... Network centralization measures

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

Variational approach: The other approach is based on the variance. First we compute the average node centrality

$$\bar{p} = \frac{1}{n} \sum_{v \in \mathcal{V}} p(v)$$

and then define

$$V_p(\mathcal{N}) = \frac{1}{n} \sum_{v \in \mathcal{V}} (p(v) - \bar{p})^2$$



Important nodes in igraph

Connectivity

V. Batagelj

Connectivity

Condensation

Bow-tie

Other
connectivities

Important
nodes

Closeness

Betweenness

Hubs and
authorities

Clustering

```
> R <- read.graph("./nets/class.net",format="pajek")
> vertex_attr(R)$shape <- NULL
> b <- betweenness(R,normalized=TRUE)
> plot(R,vertex.size=b*100)
> c <- closeness(R,normalized=TRUE)
> plot(R,vertex.size=c*100)
> e <- eigen_centrality(R)
> plot(R,vertex.size=e$vector*30)
> hub=hub.score(R)$vector
> plot(R,vertex.size=hub*20)
> aut=authority.score(R)$vector
> plot(R,vertex.size=aut*20)
> b <- bonpow(R,rescale=TRUE)
> plot(R,vertex.size=b*200)
> # clustering coefficient
> t <- transitivity(R,type="local")
> plot(R,vertex.size=t*25)
```



Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

Introduction to Network Analysis using **Pajek**

5. Structure of networks: cohesion

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

PhD and MS program in Statistics
University of Ljubljana, 2022



Outline

Cohesion

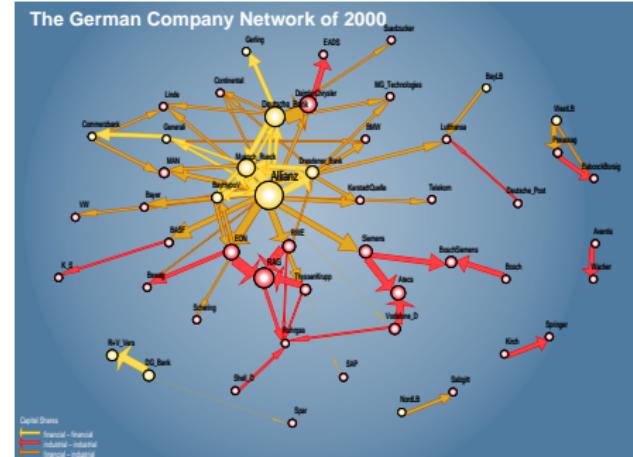
V. Batagelj

Islands

Cores

Generalized cores

- 1 Islands
- 2 Cores
- 3 Generalized cores



L. Kreml, MPI.

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

Current version of slides (March 23, 2022 at 00 : 19): [slides PDF](#)



Islands

Cohesion

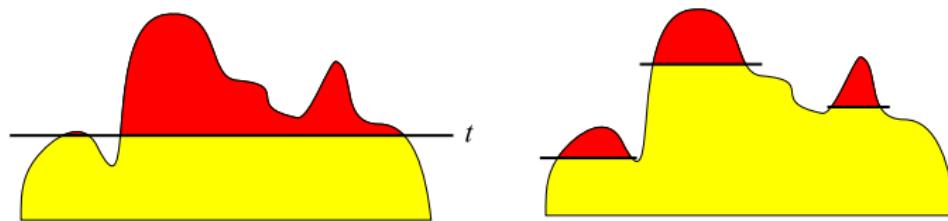
V. Batagelj

Islands

Cores

Generalized
cores

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

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

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.

The set of nodes $\mathcal{C} \subseteq \mathcal{V}$ is a **local node peak**, if it is a regular node island and all of its nodes have the same value. Node island with a single local node peak is called a **simple node island**. In similar way we define simple link island.



... Islands

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

A set of nodes $C \subseteq \mathcal{V}$ is a *regular node island* in network

$\mathcal{N} = (\mathcal{V}, \mathcal{L}, p)$, $p : \mathcal{V} \rightarrow \mathbb{R}$ 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

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

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

Network/Create Partition/Islands/Line Weights
Operations/Network+Vector/Islands/Vertex
Property



Some properties of node islands

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

- The sets of nodes of connected components of node-cut at selected level t are regular node islands.
- The set $\mathcal{H}_p(\mathcal{N})$ of all regular node islands of network \mathcal{N} is a complete hierarchy:
 - two islands are disjoint or one of them is a subset of the other
 - each node belongs to at least one island
- Node islands are invariant for the strictly increasing transformations of the property p .
- Two linked nodes cannot belong to two disjoint regular node islands.



Simple node islands

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

- The set of nodes $\mathcal{C} \subseteq \mathcal{V}$ is a **local node peak**, if it is a regular node island and all of its nodes have the same value.
- Node island with a single local node peak is called a **simple node island**.
- The types of node islands:
 - FLAT – all nodes have the same value
 - SINGLE – island has a single local node peak
 - MULTI – island has more than one local node peaks
- Only the islands of type FLAT or SINGLE are simple islands.



Some properties of link islands

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

- The sets of nodes of connected components of link-cut at selected level t are regular link islands.
- The set $\mathcal{H}_w(\mathcal{N})$ of all nondegenerated regular link islands of network \mathcal{N} is hierarchy (not necessarily complete):
 - two islands are disjoint or one of them is a subset of the other
 - Link islands are invariant for the strictly increasing transformations of the weight w .
 - Two linked nodes may belong to two disjoint regular link islands.



Simple link islands

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

- The set of nodes $\mathcal{C} \subseteq \mathcal{V}$ is a **local link peak**, if it is a regular link island and there exists a spanning tree of the corresponding induced network, in which all links have the same value as the link with the largest value.
- Link island with a single local link peak is called a **simple link island**.
- The types of link islands:
 - FLAT – there exists a spanning tree, in which all links have the same value as the link with the largest value.
 - SINGLE – island has a single local link peak.
 - MULTI – island has more than one local link peaks.
- Only the islands of type FLAT or SINGLE are simple islands.



US patents

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

US patents network ([Nber, US Patents](#)) has 3774768 nodes and 16522438 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. We determined all (2,90)-islands. The corresponding subnetwork has 470137 nodes, 307472 arcs and for different k : $C_2 = 187610$, $C_5 = 8859$, $C_{30} = 101$, $C_{50} = 30$, ... 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



Distribution of island size

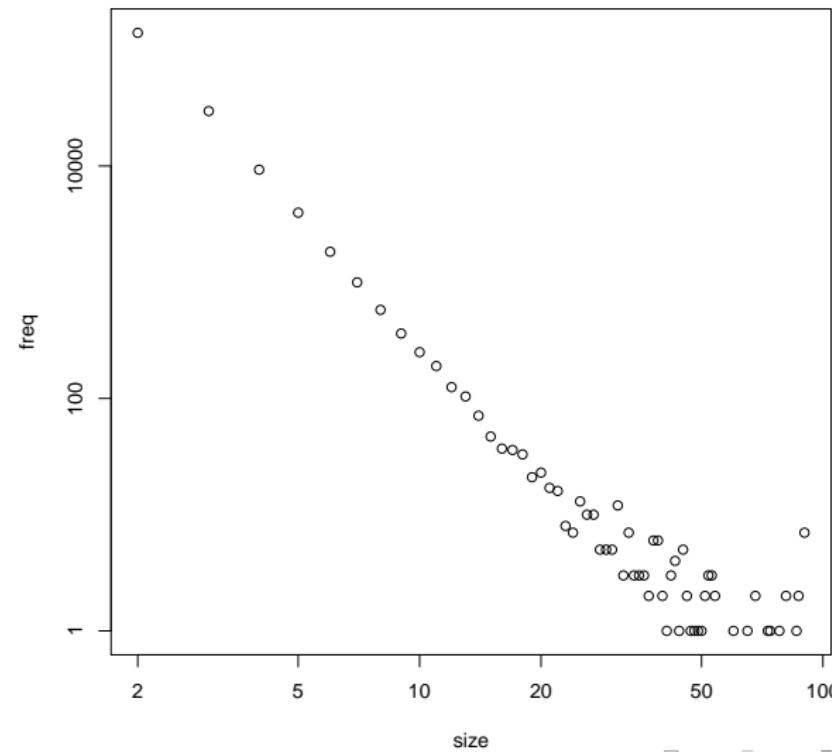
Cohesion

V. Batagelj

Islands

Cores

Generalized
cores





Main path and main island in US Patents

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

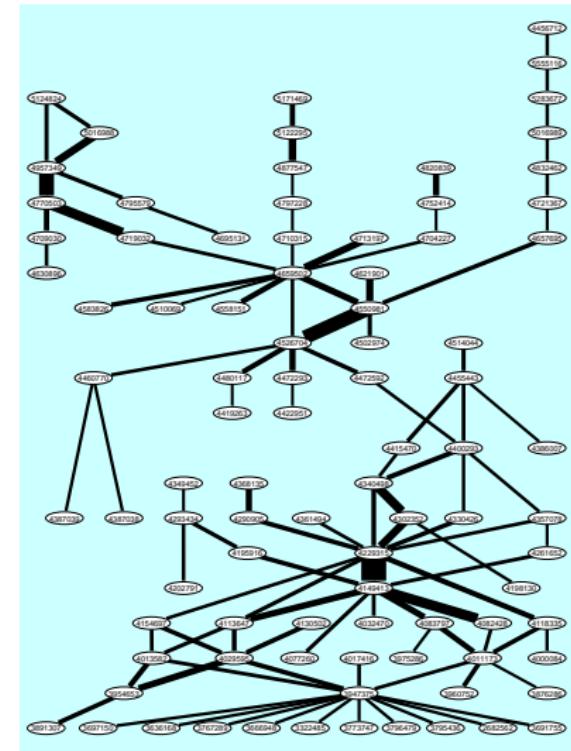
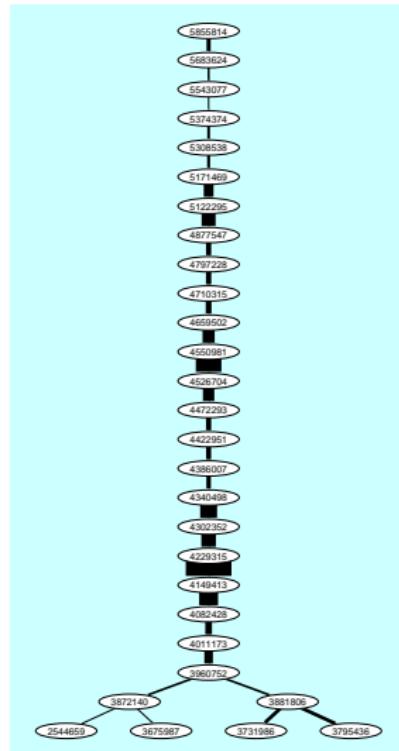
Cohesion

V. Batagelj

Islands

Cores

Generalized cores





Main island – Liquid crystal display

Cohesion

V. Batagelj

Islands

Cores

Generalized cores

Table 1: Patents on the liquid-crystal display

patient	filed (X)	date (Y)	inventor(s) and title
3005262	Jan 29, 1952	Apr 12, 1953	Process for preparing short and the low and the formation and use thereof
3324285	May 30, 1952	Wunder, et al. Reduction of aromatic carboxylic acids by means of a substituted organic ester	
3324286	May 30, 1952	Process for the preparation of substituted organic ester	
3636108	Jan 18, 1972	Gray, et al. Preparation of polymer-like aromatic thermotropic compounds	
3636109	May 30, 1972	Gray, et al. Liquid crystal thermal imaging crystals having an undistorted image on a disturbed background	
3671987	Jul 11, 1972	Gray, et al. Liquid crystal thermal imaging crystals having an undistorted image on a disturbed background	
3691755	Sep 19, 1972	Gray, et al. Liquid crystal thermal imaging crystals having an undistorted image on a disturbed background	
3691756	Oct 10, 1972	Gray, et al. Liquid crystal thermal imaging crystals having an undistorted image on a disturbed background	
3731986	May 8, 1973	Ferguson, et al. Display device utilizing liquid crystal light	
3767289	Oct 23, 1973	Arivian, et al. Class of stable trans-esther compounds, their preparation and method of use in electro-optical devices and others in a range up to 100°C	
3774247	Nov 20, 1973	Stolzenhaar, et al. Nematic liquid crystal compositions	
3795436	Mar 5, 1974	Stolzenhaar, et al. Nematic liquid crystal compositions which exhibit the Kerr effect at isotropic temperatures	
3796749	Mar 12, 1974	Stolzenhaar, et al. Nematic liquid-nematic cell utilizing a semicapillary material which exhibits the Kerr effect at isotropic temperatures	
3877440	Mar 18, 1974	Stolzenhaar, et al. Nematic liquid-crystalline compositions and method	
3878284	Aug 8, 1974	Dietrich, et al. Use of nematic liquid-crystalline substances in electro-optical display devices	
3881868	Mar 6, 1975	Tsukamoto, et al. Phase control of the voltages applied to the electrodes for a cholesteric to nematic phase transition display	
3947375	Mar 20, 1975	Gray, et al. Liquid crystal materials and devices	
3952150	Mar 20, 1975	Gray, et al. Liquid crystal materials and devices having high dielectric anisotropy and display device incorporating same	
3962572	Jan 1, 1976	Klumperman, et al. Liquid crystal compositions containing cyanobiphenyls and method of synthesis	
3975286	Aug 17, 1976	Gray, et al. Liquid crystal materials and devices incorporating them	
4000694	Dec 26, 1976	Gray, et al. Liquid crystal materials for electro-optical display devices	
4011173	Mar 8, 1977	Stolzenhaar, Modified nematic mixture with	
4031582	Mar 22, 1977	Gericke, et al. Liquid crystal compounds and electro-optic devices incorporating them	
4077446	Apr 12, 1977	Gray, et al. Process for preparing some liquid and liquid-crystal compositions and method of synthesis	
4025095	Jan 14, 1977	Rose, et al. Novel liquid crystal compounds and electro-optic devices incorporating them	
4032478	Jan 28, 1977	Gray, et al. Liquid crystal materials and devices	
4077260	Mar 7, 1978	Gray, et al. Optically active cyano-biphenyl compounds and liquid crystal materials containing them	
4082428	Apr 4, 1978	Gray, et al. Liquid crystal composition and method	

Table 2: Patents on the liquid-crystal display

patient	date (X)	inventor(s) and title
4113647	Apr 12, 1978	Costes, et al. Liquid crystalline materials
4118320	Dec 3, 1978	Costes, et al. Liquid crystalline materials of reduced viscosity
4143622	Apr 17, 1979	Gray, et al. Liquid crystal materials having a benzene derivative
4149412	Apr 17, 1979	Gray, et al. Optically active liquid crystal mixture and liquid crystal diesters containing thioether
4154007	May 15, 1979	Gray, et al. Liquid crystal materials having hydroxyphenyl derivatives
4161820	Aug 1, 1979	Boller, et al. Liquid crystal mixtures
4202791	Oct 12, 1979	Sato, et al. Novel liquid crystal materials
4221152	Oct 12, 1979	Gray, et al. Liquid crystal materials having a benzene derivative
4261630	Apr 14, 1980	Gray, et al. Liquid crystal compounds and materials and devices containing them
4289055	Sep 22, 1980	Gray, et al. Electro-optic liquid crystal compounds
4293434	Oct 6, 1980	Dietrich, et al. Liquid crystal compounds
4302322	Nov 24, 1980	Dietrich, et al. Liquid crystal materials and devices, the preparation and their use as components of liquid crystal diodes
4308404	Jul 20, 1982	Eldenbach, et al. Cyclohexyl biphenyl, their preparations and uses
4319425	Sep 14, 1982	Sugimoto, et al. Halogenated ether derivatives
4325076	Nov 2, 1982	Caro, et al. Liquid crystal compounds containing an acyclic ring and exhibiting a low dielectric anisotropy and liquid crystal materials containing them
4326155	Nov 20, 1982	Ozawa, et al. Anisotropic cyclohexyl cyclohexyl ethers
4366008	May 31, 1984	Krasne, et al. Liquid crystalline naphthalene derivatives
4370708	Jun 7, 1984	Pulak, et al. (Trans-4-(2,4-dicyanophenoxy)-benzoic acid -cyano-alkoxy)cyanobiphenyl
4376709	Jun 7, 1984	Sugimoto, et al. (Trans-4-(trans-4-(2,4-dicyanophenoxy)-cyclohexane -cyano-alkoxy)cyanobiphenyl
4406252	Aug 23, 1985	Eldebach, et al. Liquid crystalline cyclohexylphenyl derivatives
4415479	Nov 15, 1985	Sugimoto, et al. Liquid crystalline cyanobiphenyls and electro-optical display elements based thereon
4419303	Dec 6, 1986	Caro, et al. Liquid crystalline cyclohexylbenzonitrile derivatives
4422951	Dec 27, 1986	Sugimoto, et al. Liquid crystal benzene derivatives
4454172	Jan 26, 1987	Christie, et al. Bisimide-like triazine composition
4457612	Jan 26, 1987	Petrzik, et al. High temperature liquid crystal substances of four rings and liquid crystal compositions containing the same
4467203	Feb 18, 1988	Takatori, et al. Nematic liquid crystal compositions
4472202	Feb 18, 1988	Sugimoto, et al. High temperature liquid-crystalline ester
4472202	Sep 18, 1988	Eldebach, et al. Liquid crystal devices having adjacent electrode terminals set equal in length
4480117	Oct 20, 1988	Matsuura, et al. Liquid crystal composition and liquid crystal display elements
4502974	Mar 5, 1989	Eldebach, et al. Cyclohexane derivatives
4510008	Apr 9, 1989	Eldebach, et al. Cyclohexane derivatives

Table 3: Patents on the liquid-crystal display

patient	date (X)	inventor(s) and title
4514004	Apr 30, 1989	Trans-4-(4-phenylcyclohexyl)-2-vinyl-4-(p-substituted phenyl) cyclohexylbenzene and liquid crystal mixture
4526704	Jul 2, 1989	Petrzik, et al. Multiring liquid crystal esters
4530454	Nov 28, 1989	Eldebach, et al. Liquid crystal materials having a pentamethylcyclopentadiene
4558153	Dec 10, 1989	Takatori, et al. Nematic liquid crystal compounds
4561253	Dec 11, 1989	Petrzik, et al. Planar liquid crystal mixtures
4589096	Dec 23, 1989	Petrzik, et al. Benzene derivatives
4600002	Dec 26, 1989	Petrzik, et al. Cyclohexane derivatives
4605502	Apr 21, 1990	Fournet, et al. Ethane derivatives
4605131	Sep 22, 1990	Bakewell, et al. Disubstituted ethanes and their use in liquid crystal displays
4614227	Nov 3, 1990	Krasne, et al. Liquid crystal compound
4627047	Dec 18, 1990	Eldebach, et al. Asymmetric liquid crystal mixtures and liquid crystal mixtures therewith
4713307	Dec 15, 1991	Struktur, et al. Structure-containing heterocyclic compounds
4718032	Jan 12, 1992	Wiedeker, et al. Cyclohexane derivatives
4718367	Jan 28, 1992	Yoshimura, et al. Liquid crystal materials having structure-containing heterocyclic compounds
4737003	Feb 12, 1992	Eldebach, et al. Structure-containing heterocyclic compounds
4770053	Sep 13, 1992	Wiedeker, et al. Liquid crystalline compounds
4795579	Oct 20, 1992	Eldebach, et al., et al. 2-(4-phenyl-4-hydroxy-4-phenyl)biphenyl and their derivatives, their production, and their use in liquid crystal display devices
4797229	Jan 18, 1993	Krasne, et al. Liquid crystal composition and liquid crystal composition containing same
4820620	Apr 16, 1993	Krasne, et al. Structure-containing heterocyclic compounds
4823651	Dec 21, 1993	Schad, et al. Asymmetric liquid crystal mixtures and liquid crystal mixtures therewith
4873747	Oct 21, 1993	Wiedeker, et al. Structure-containing heterocyclic compounds
4957349	Sep 28, 1996	Eldebach, et al. Structure-containing heterocyclic compounds
5010088	May 21, 1999	Eldebach, et al. Liquid crystal element with improved contrast and a brightness
5010089	May 21, 1999	Okada, Liquid crystal element with improved contrast and said access
5122295	Jun 16, 2000	Wohrer, et al. Matte liquid crystal display
5124824	Jun 23, 2000	Wohrer, et al. Matte liquid crystal display device with a birefringent compensator
5171409	Dec 15, 2000	Sugino, et al. Liquid crystal composition and liquid crystal composition containing same
5203627	Feb 1, 2001	Sugino, et al. Liquid crystal display with ground regions
5247744	Jul 21, 2001	Wohrer, et al. Structure-containing heterocyclic compounds
5253295	Aug 21, 2001	Wohrer, et al. Structure-containing heterocyclic compounds
5266386	Sep 18, 2001	Wohrer, et al. Superwet liquid-crystal display
5274374	Dec 20, 2001	Wohrer, et al. Structure-containing heterocyclic compounds
5277545	Jan 22, 2002	Wohrer, et al. Structure-containing heterocyclic compounds
5295516	Sep 10, 2002	Eldebach, et al. Liquid crystal display having adjacent electrode terminals set equal in length
5303224	Nov 4, 2002	Eldebach, et al. Liquid crystal composition and liquid crystal display elements
5356514	Jan 5, 2003	Matsuura, et al. Liquid crystal composition and liquid crystal display elements

V. Batagelj

Cohesion



The Edinburgh Associative Thesaurus

$n = 23219$, $m = 325624$, transitivity weight

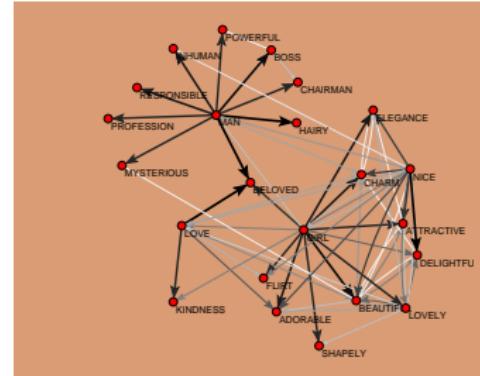
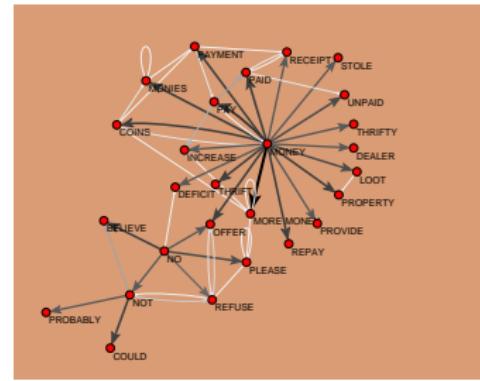
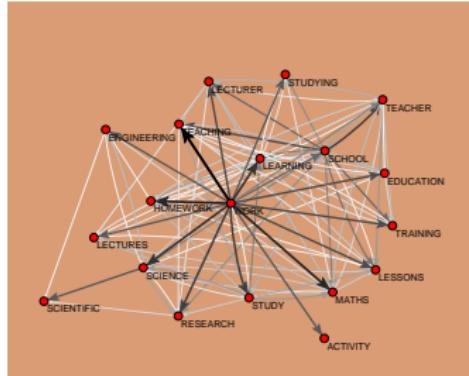
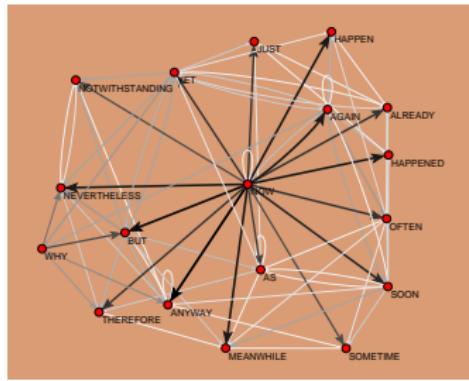
Cohesion

V. Batagelj

Islands

Cores

Generalized
cores





Dense groups

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

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 difficult to determine.



Cores and generalized cores

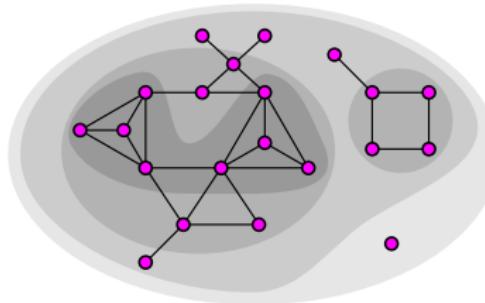
Cohesion

V. Batagelj

Islands

Cores

Generalized
cores



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

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

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

- The cores are nested: $i < j \implies \mathcal{H}_j \subseteq \mathcal{H}_i$
- 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 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we recursively delete all nodes, and edges incident with them, of degree less than k , the remaining graph is the k -core.



6-core of Krebs Internet industries

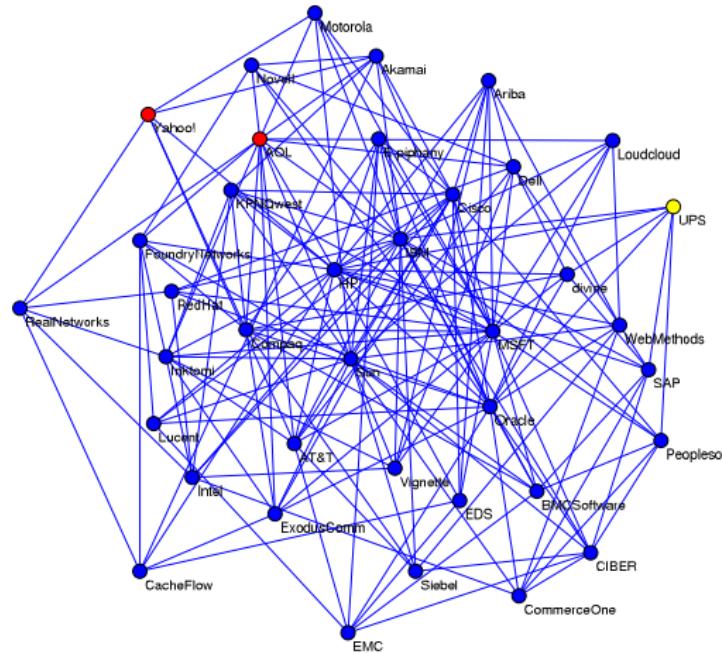
Cohesion

V. Batagelj

Islands

Cores

Generalized cores





Generalized cores

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

The notion of core can be generalized to networks. Let $\mathcal{N} = (\mathcal{V}, \mathcal{E}, w)$ be a network, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph and $w : \mathcal{E} \rightarrow \mathbb{R}$ 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:

$$p_S(v, U) = \sum_{u \in N_U(v)} w(v, u), \text{ where } w : \mathcal{E} \rightarrow \mathbb{R}_0^+$$

$$p_M(v, U) = \max_{u \in N_U(v)} w(v, u), \text{ where } w : \mathcal{E} \rightarrow \mathbb{R}$$

$$p_t(v, U) = \frac{|\mathcal{L}(U) \cap \mathcal{L}(K(N^+(v)))|}{|\mathcal{L}(K(N^+(v)))|}$$

$$p_k(v, U) = \text{number of cycles of length } k \text{ through node } v \text{ in } (U, \mathcal{E}|U)$$

The subgraph $\mathcal{H} = (C, \mathcal{E}|C)$ 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

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

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

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

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

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

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 $O(\deg_C(v))$.

For details see the [paper](#).



Cores and generalized cores / Pajek commands

Cohesion

V. Batagelj

Islands

Cores

Generalized
cores

```
File/Network/Read [Geom.net]
Network/Create Partition/k-Core/All
Info/Partition
Operations/Network+Partition/Extract Subnetwork [13-*]
Draw/Network+First Partition
Layout/Energy/Kamada-Kawai
Options/Values of lines/Similarities
Layout/Energy/Kamada-Kawai
Operations/Network+Partition/Extract Subnetwork [21]
Draw/Network
Layout/Energy/Kamada-Kawai
Options/Values of lines/Forget
Layout/Energy/Kamada-Kawai
[select Geom.net]
Network/Create Vector/Generalized Cores/Sum/All
Info/Vector
Vector/Make Partition/by Intervals/Selected Thresholds [4]
Info/Partition
Operations/Network+Partition/Extract Subnetwork [2]
Draw/Network
Options/Values of lines/Similarities
Layout/Energy/Fruchterman-Reingold
```



Cores of orders 10–21 in Computational Geometry

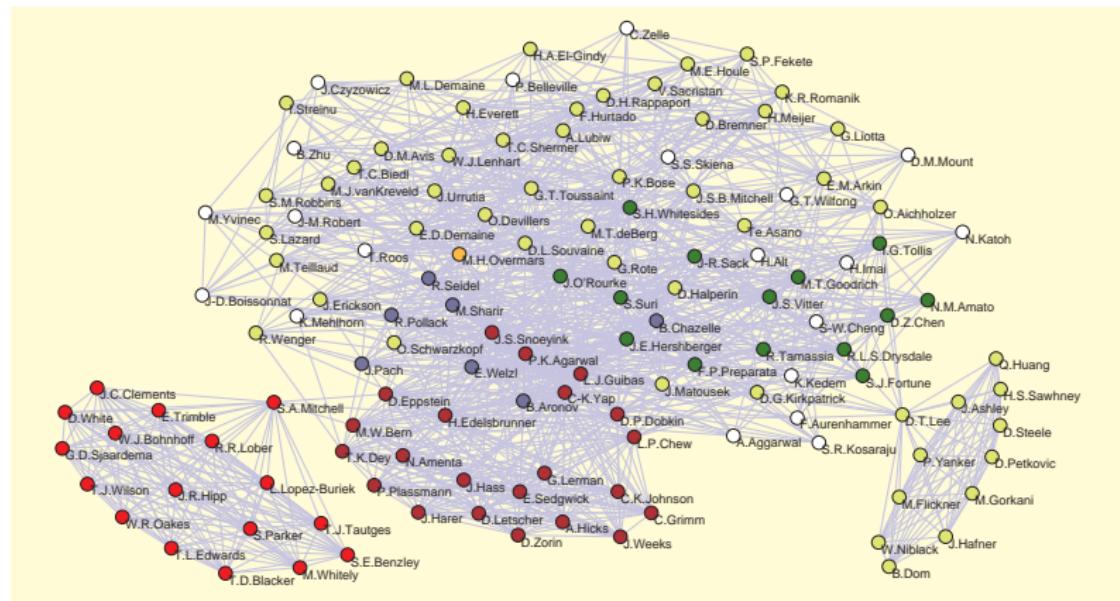
Cohesion

V. Batagelj

Islands

Cores

Generalized cores





p_S -core at level 46 of Geombib network

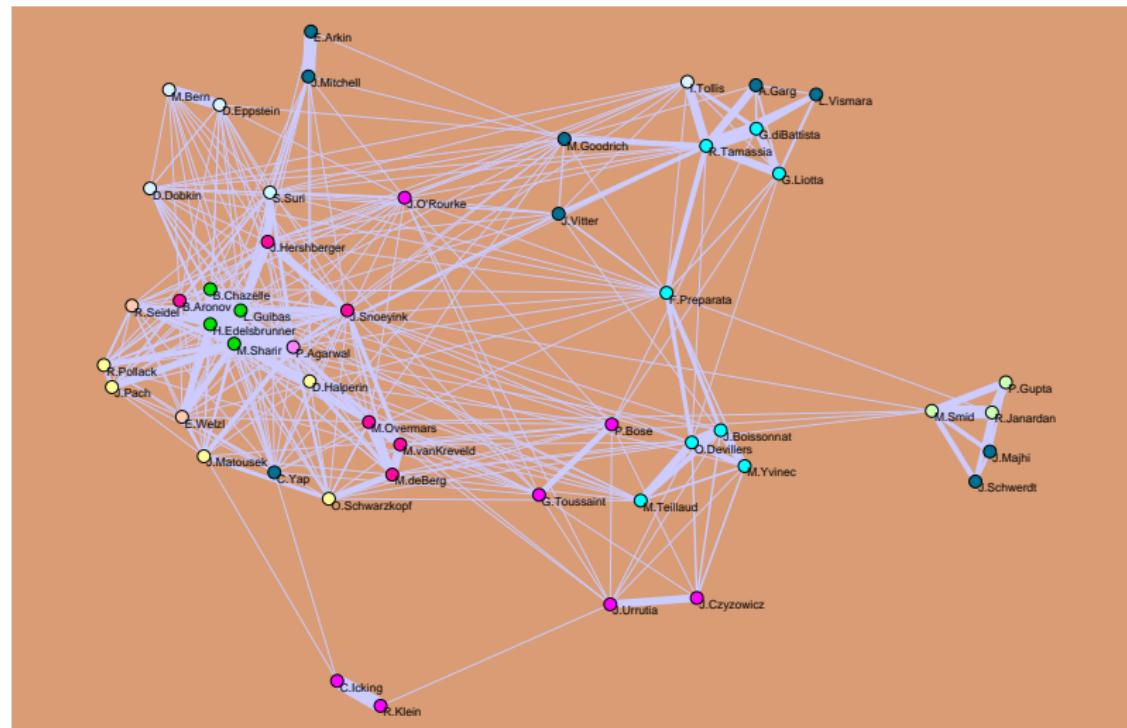
Cohesion

V. Batagelj

Islands

Cores

Generalized cores





Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

Introduction to Network Analysis using **Pajek**

6. Structure of networks: Acyclic networks and patterns search

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

PhD and MS program in Statistics
University of Ljubljana, 2022



Outline

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

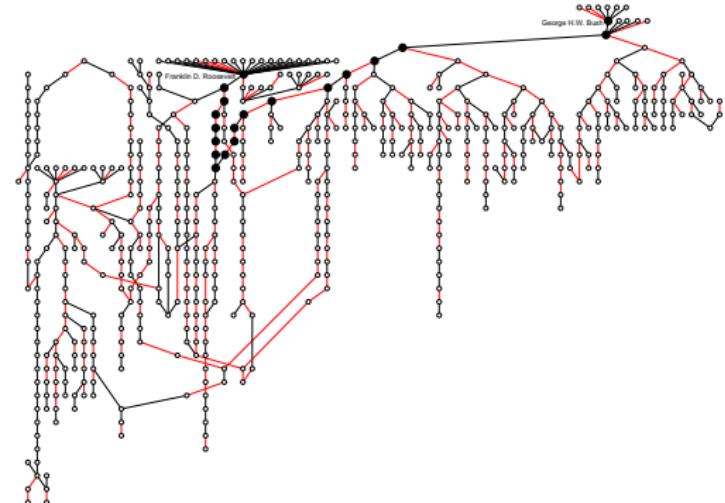
Pattern
searching

Triads

Motifs

Graphlets

- 1 Acyclic networks
- 2 Numberings
- 3 Citation networks
- 4 Genealogies
- 5 Pattern searching
- 6 Triads
- 7 Motifs
- 8 Graphlets



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

Current version of slides (March 23, 2022 at 00 : 25): [slides PDF](#)



Acyclic networks

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

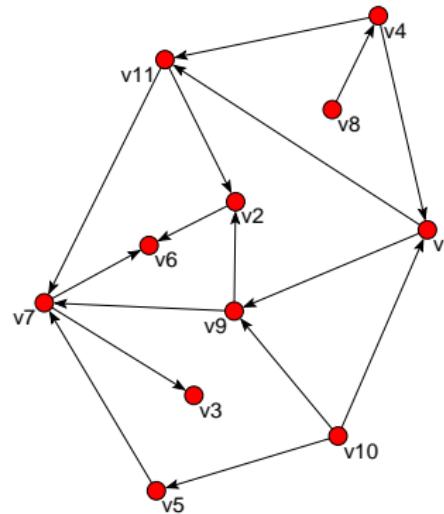
Genealogies

Pattern
searching

Triads

Motifs

Graphlets



acyclic.paj

Network $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, $\mathcal{R} \subseteq \mathcal{V} \times \mathcal{V}$ is **acyclic**, if it doesn't contain any (proper) cycle.

$$\overline{\mathcal{R}} \cap I = \emptyset$$

In some cases we allow loops.
Examples: citation networks,
genealogies, project networks,
...

In real-life acyclic networks we usually have a node property $p : \mathcal{V} \rightarrow \mathbb{R}$ (most often time), that is **compatible** with arcs

$$(u, v) \in \mathcal{R} \Rightarrow p(u) < p(v)$$

Network/Create Partition/Components/Strong [2] ↗ ↘ ↙ ↛ ↜ ↝ ↞ ↞ ↞



Basic properties of acyclic networks

Let $\mathcal{G} = (\mathcal{V}, R)$ be acyclic and $\mathcal{U} \subseteq \mathcal{V}$, then $\mathcal{G}|_{\mathcal{U}} = (\mathcal{U}, R|_{\mathcal{U}})$, $R|_{\mathcal{U}} = R \cap \mathcal{U} \times \mathcal{U}$ is also acyclic.

Let $\mathcal{G} = (\mathcal{V}, R)$ be acyclic, then $\mathcal{G}' = (\mathcal{V}, R^{-1})$ is also acyclic.
Duality.

The set of *sources* $\text{Min}_R(\mathcal{V}) = \{v : \neg \exists u \in \mathcal{V} : (u, v) \in R\}$ and the set of *sinks* $\text{Max}_R(\mathcal{V}) = \{v : \neg \exists u \in \mathcal{V} : (v, u) \in R\}$ are nonempty (in finite networks).

Transitive closure \overline{R} of an acyclic relation R is acyclic.

Relation Q is a *skeleton* of relation R iff $Q \subseteq R$, $\overline{Q} = \overline{R}$ and relation Q is minimal such relation – no arc can be deleted from it without destroying the second property.

A general relation (graph) can have several skeletons; but in a case of acyclic relation it is uniquely determined $Q = R \setminus R * \overline{R}$.



Depth

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

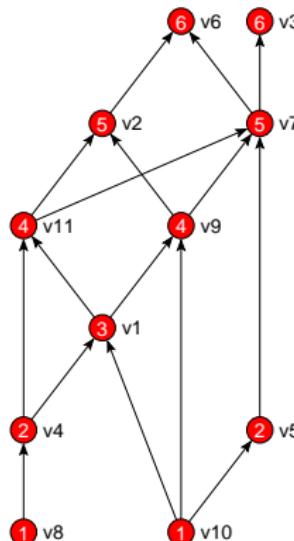
Genealogies

Pattern
searching

Triads

Motifs

Graphlets



Mapping $h : \mathcal{V} \rightarrow \mathbb{N}^+$ is called *depth* or *level* if all differences on the longest path and the initial value equal to 1.

```
 $\mathcal{U} \leftarrow \mathcal{V}; k \leftarrow 0$ 
while  $\mathcal{U} \neq \emptyset$  do
     $\mathcal{T} \leftarrow \text{Min}_R(\mathcal{U}); k \leftarrow k + 1$ 
    for  $v \in \mathcal{T}$  do  $h(v) \leftarrow k$ 
     $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{T}$ 
```

Drawing on levels. Macro Layers.



p-graph of Bouchard's genealogy

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

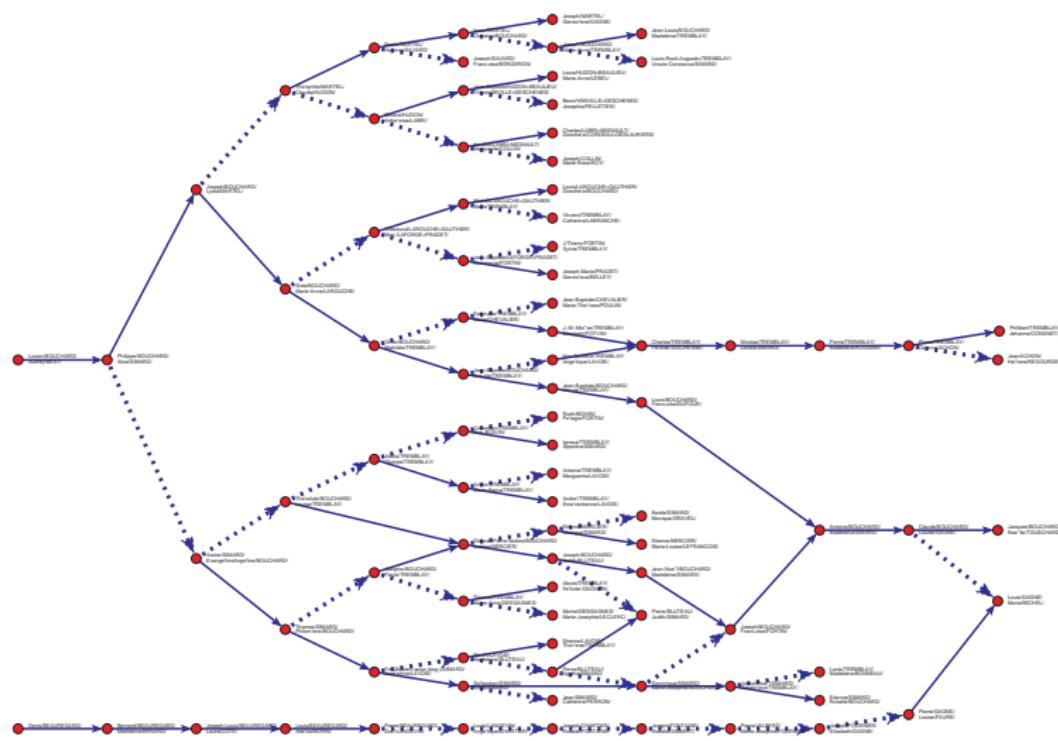
Genealogies

Pattern
searching

Triads

Motifs

Graphlets





Topological numberings

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

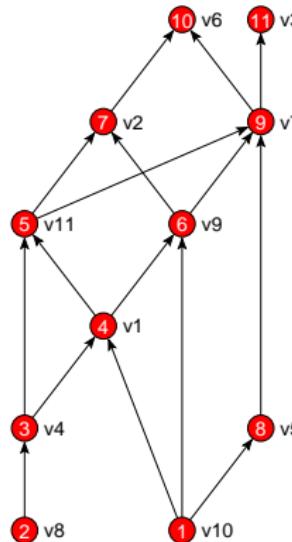
Genealogies

Pattern
searching

Triads

Motifs

Graphlets



Injective mapping $h : \mathcal{V} \rightarrow 1..|\mathcal{V}|$ compatible with relation R is called a **topological numbering**.
'Topological sort'

```
 $\mathcal{U} \leftarrow \mathcal{V}; k \leftarrow 0$ 
while  $\mathcal{U} \neq \emptyset$  do
    select  $v \in \text{Min}_R(\mathcal{U})$ ;  $k \leftarrow k + 1$ 
     $h(v) \leftarrow k$ 
     $\mathcal{U} \leftarrow \mathcal{U} \setminus \{v\}$ 
```

Matrix display of acyclic network with vertices reordered according to a topological numbering has a zero lower triangle.



... Topological numberings

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

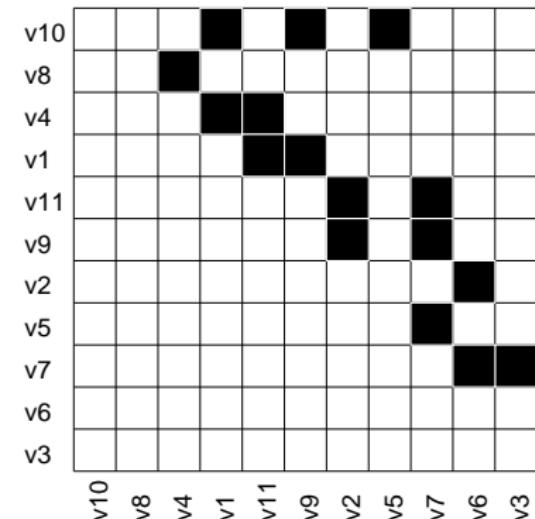
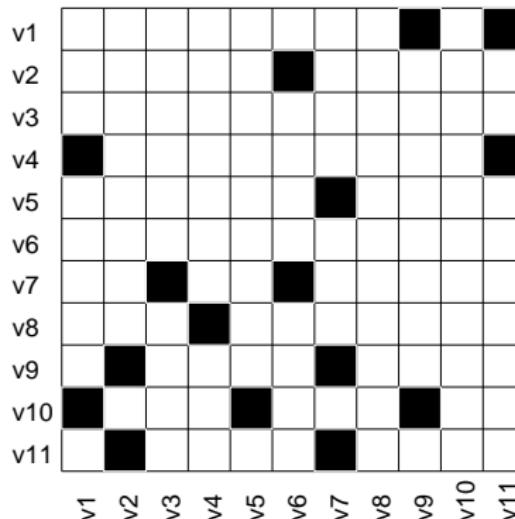
Genealogies

Pattern
searching

Triads

Motifs

Graphlets



```
read or select [Acyclic.paj]
Network/Acyclic Network/Depth Partition/Acyclic
Partition/Make Permutation
File/Network/Export as Matrix to EPS/Using Permutation [acy.eps]
```



Topological numberings and functions on acyclic networks

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

Let the function $f : \mathcal{V} \rightarrow \mathbb{R}$ be defined in the following way:

- $f(v)$ is knownn in sources $v \in \text{Min}_R(\mathcal{V})$
- $f(v) = F(\{f(u) : uRv\})$

If we compute the values of function f in a sequence determined by a topological numbering we can compute them in one pass since for each node $v \in \mathcal{V}$ the values of f needed for its computation are already known.



Topological numberings – CPM

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

CPM (Critical Path Method): A project consists of tasks. Nodes of a project network represent states of the project and arcs represent tasks. Every project network is acyclic. For each task (u, v) its execution time $t(u, v)$ is known. A task can start only when all the preceding tasks are finished. We want to know what is the shortest time in which the project can be completed.

Let $T(v)$ denotes the earliest time of completion of all tasks entering the state v .

$$T(v) = 0, \quad v \in \text{Min}_R(\mathcal{V})$$

$$T(v) = \max_{u: u R v} (T(u) + t(u, v))$$

Network/Acyclic Network/Critical Path Method–CPM



Citation networks

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

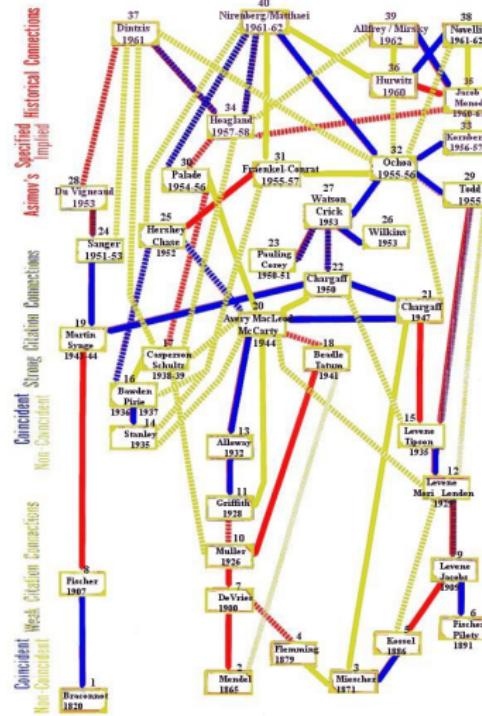
Genealogies

Pattern
searching

Triads

Motifs

Graphlets



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 provide us with automatic way to identify the (most) important part of the citation network. For two of these indices we developed algorithms to efficiently compute them.



... Citation networks

In a given set of units/nodes \mathcal{U} (articles, books, works, etc.) we introduce a *citing relation*/set of arcs $R \subseteq \mathcal{U} \times \mathcal{U}$

$$uRv \equiv u \text{ cites } v$$

which determines a *citation network* $\mathcal{N} = (\mathcal{U}, R)$.

A citing relation is usually *irreflexive* (no loops) and (almost) *acyclic*. We shall assume that it has these two properties. Since in real-life citation networks the strong components are small (usually 2 or 3 nodes) we can transform such network into an acyclic network by shrinking strong components and deleting loops. Other approaches exist. It is also useful to transform a citation network to its *standardized* form by adding a common *source* node $s \notin \mathcal{U}$ and a common *sink* node $t \notin \mathcal{U}$. The source s is linked by an arc to all minimal elements of R ; and all maximal elements of R are linked to the sink t . We add also the ‘feedback’ arc (t, s) .



Search path count method

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

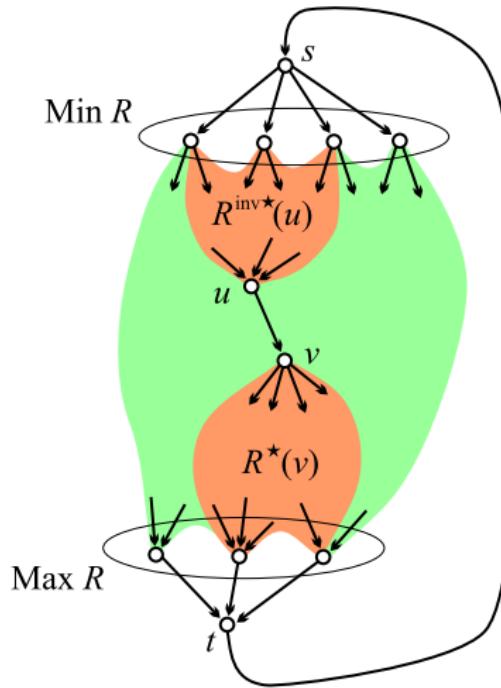
Genealogies

Pattern
searching

Triads

Motifs

Graphlets



The *search path count* (SPC) method is based on counters $n(u, v)$ that count the number of different paths from s to t through the arc (u, v) . To compute $n(u, v)$ we introduce two auxiliary quantities: $n^-(v)$ counts the number of different paths from s to v , and $n^+(v)$ counts the number of different paths from v to t .



Fast algorithm for SPC

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

It follows by basic principles of combinatorics that

$$n(u, v) = n^-(u) \cdot n^+(v), \quad (u, v) \in R$$

where

$$n^-(u) = \begin{cases} 1 & u = s \\ \sum_{v: v \mathbf{R} u} n^-(v) & \text{otherwise} \end{cases}$$

and

$$n^+(u) = \begin{cases} 1 & u = t \\ \sum_{v: u \mathbf{R} v} n^+(v) & \text{otherwise} \end{cases}$$

This is the basis of an efficient algorithm for computing $n(u, v)$ – after the topological sort of the graph we can compute, using the above relations in topological order, the weights in time of order $O(m)$, $m = |R|$. The topological order ensures that all the quantities in the right sides of the above equalities are already computed when needed.



Hummon and Doreian indices and SPC

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

The Hummon and Doreian indices are defined as follows:

- **search path link count** (SPLC) method: $w_l(u, v)$ equals the number of “*all possible search paths through the network emanating from an origin node*” through the arc $(u, v) \in R$.
- **search path node pair** (SPNP) method: $w_p(u, v)$ “*accounts for all connected node pairs along the paths through the arc* $(u, v) \in R$ ”.

We get the SPLC weights by applying the SPC method on the network obtained from a given standardized network by linking the source s by an arc to each nonminimal vertex from \mathcal{U} ; and the SPNP weights by applying the SPC method on the network obtained from the SPLC network by additionally linking by an arc each nonmaximal vertex from \mathcal{U} to the sink t .



Node weights

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

The quantities used to compute the arc weights w can be used also to define the corresponding node weights t

$$t_c(u) = n^-(u) \cdot n^+(u)$$

$$t_l(u) = n_l^-(u) \cdot n_l^+(u)$$

$$t_p(u) = n_p^-(u) \cdot n_p^+(u)$$

They are counting the number of paths of selected type through the node u .

Network/Acyclic Network/Citation Weights



Properties of SPC weights

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

The values of counters $n(u, v)$ form a flow in the citation network – the *Kirchoff's node law* holds: For every node u in a standardized citation network $\text{incoming flow} = \text{outgoing flow}$:

$$\sum_{v:vRu} n(v, u) = \sum_{v:uRv} n(u, v) = n^-(u) \cdot n^+(u)$$

The weight $n(t, s)$ equals to the total flow through network and provides a natural normalization of weights

$$w(u, v) = \frac{n(u, v)}{n(t, s)} \quad \Rightarrow \quad 0 \leq w(u, v) \leq 1$$

and if C is a minimal arc-cut-set $\sum_{(u,v) \in C} w(u, v) = 1$.

In large networks the values of weights can grow very large. This should be considered in the implementation of the algorithms.



Nonacyclic citation networks

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

If there is a cycle in a network then there is also an infinite number of trails between some units. There are some standard approaches to overcome the problem: to introduce some 'aging' factor which makes the total weight of all trails converge to some finite value; or to restrict the definition of a weight to some finite subset of trails – for example paths or geodesics. But, new problems arise: What is the right value of the 'aging' factor? Is there an efficient algorithm to count the restricted trails?

The other possibility, since a citation network is usually almost acyclic, is to transform it into an acyclic network

- by identification (shrinking) of cyclic groups (nontrivial strong components), or
- by deleting some arcs, or
- by transformations such as the 'preprint' transformation.



Preprint transformation

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

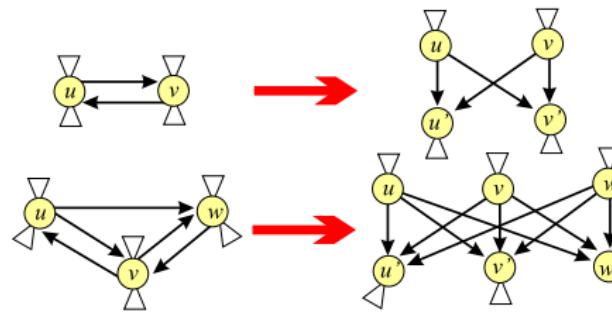
Genealogies

Pattern
searching

Triads

Motifs

Graphlets



The *preprint transformation* is based on the following idea: Each paper from a strong component is duplicated with its 'preprint' version. The papers inside strong component cite preprints.

Large strong components in citation network are unlikely – their presence usually indicates an error in the data.

An exception from this rule is the **HEP** citation network of High Energy Particle Physics literature from **arXiv**. In it different versions of the same paper are treated as a unit. This leads to large strongly connected components. The idea of preprint transformation could be used also in this case to eliminate cycles.



Probabilistic flow

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

Another way to measure the importance of nodes and arcs in acyclic networks is the following. Let $\mathcal{N} = (\mathcal{V}, \mathcal{A})$ be a standardized acyclic network with source $s \in \mathcal{V}$ and sink $t \in \mathcal{V}$. The *node potential*, $p(v)$, is defined by

$$p(s) = 1 \quad \text{and} \quad p(v) = \sum_{u:(u,v) \in \mathcal{A}} \frac{p(u)}{\text{outdeg}(u)}$$

The flow on the arc (u, v) is defined as $\varphi(u, v) = \frac{p(u)}{\text{outdeg}(u)}$. It follows immediately that

$$p(v) = \sum_{u:(u,v) \in \mathcal{A}} \varphi(u, v)$$

and also,

$$\sum_{u:(v,u) \in \mathcal{A}} \varphi(v, u) = \sum_{u:(v,u) \in \mathcal{A}} \frac{p(v)}{\text{outdeg}(v)} = \frac{p(v)}{\text{outdeg}(v)} \sum_{u:(v,u) \in \mathcal{A}} 1 = p(v)$$



... probabilistic flow

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

$$\sum_{u:(u,v) \in \mathcal{A}} \varphi(u, v) = \sum_{u:(v,u) \in \mathcal{A}} \varphi(v, u) = p(v)$$

which states that Kirchoff's law holds for the flow φ .

The probabilistic interpretation of flows has two parts:

- 1 The node potential of v , $p(v)$, is equal to the probability that a random walk starting in the source s goes through the node v , and
- 2 The arc flow on (u, v) , $\varphi(u, v)$, is equal to the probability that a random walk starting in the source, s , goes through the arc (u, v) .

Note that the measures p and φ consider only “users” (future) and do not depend on the past.



...probabilistic flow

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

SN5 citation network, flows multiplied with 10^6

1 BARABASI_A(1999) 286:509	2481.1796	26 SCHOUTEN_L(1993) 22:369	701.4421
2 WATTS_D(1998) 393:440	2413.1823	27 LANEMAN_J(2004) 50:3062	697.3903
3 ALBERT_R(2002) 74:47	2099.6951	28 BOYD_S(2004):	681.9335
4 WASSERMAN_S(1994) :	1807.7400	29 BARABASI_A(2004) 5:101	662.5071
5 RONAYNE_J(1987) :	1697.2066	30 AMARAL_L(2000) 97:11149	659.7386
6 WANT_R(1992) 10:91	1694.8577	31 CARZANIG_A(2001) 19:332	658.6667
7 JEONG_H(2001) 411:41	1656.5485	32 BURT_R(1992):	635.2949
8 FREEMAN_L(1979) 1:215	1559.2715	33 [ANONYMO(2008) :	621.7552
9 NEWMAN_M(2003) 45:167	1521.8437	34 JADBABAIE_A(2003) 48:988	599.2536
10 HOLBEN_B(1998) 66:1	1494.6278	35 BORGATTI_S(2002):	594.1600
11 ALBERT_R(2000) 406:378	1171.6774	36 ALBERT_R(1999) 401:130	589.2179
12 JEONG_H(2000) 407:651	1142.8359	37 NEWMAN_M(2001) 98:404	584.6247
13 FREEMAN_L(1977) 40:35	1083.6487	38 [ANONYMO(2006) :	570.9648
14 GIRVAN_M(2002) 99:7821	1055.2631	39 SHANNON_P(2003) 13:2498	566.9214
15 [ANONYMO(2011) :	940.7750	40 KATZELA_I(1996) 3:10	558.6965
16 SELBY_P(1996) 348:313	884.8034	41 LYNN_D(2008) 4:	512.6603
17 ZADEH_L(1997) 90:111	873.1572	42 BLUMENTH_D(1994) 82:1650	509.7068
18 [ANONYMO(2009) :	808.5712	43 [ANONYMO(2007) :	508.8429
19 [ANONYMO(2010) :	789.0410	44 DOROGOVT_S(2002) 51:1079	505.0996
20 STROGATZ_S(2001) 410:268	785.9130	45 SCOTT_J(2000):	497.5110
21 BOCCALET_S(2006) 424:175	782.6379	46 RAVASZ_E(2002) 297:1551	496.6379
22 GRANOVET(1973) 78:1360	777.3402	47 UETZ_P(2000) 403:623	496.3690
23 PARTON_R(1994) 127:1199	751.0084	48 ERDOS_P(1959) 6:290	483.7304
24 GAREY_M(1979) :	734.1673	49 DIAMOND_D(2008) 75:606	466.9622
25 HAGMANN_P(2008) 6:1479	718.6859	50 VANLANSCE_J(2001) 91:1574	465.9244



Genealogies

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

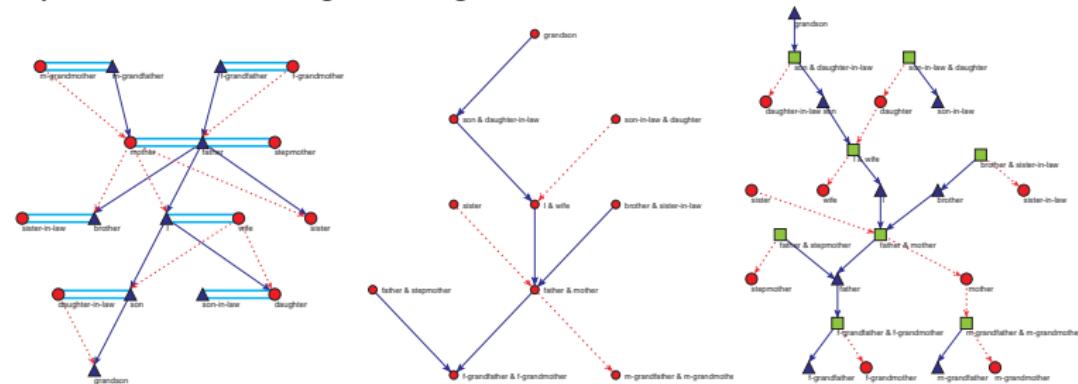
Pattern
searching

Triads

Motifs

Graphlets

Another example of acyclic networks are genealogies. In 'Sources' we already described the following network representations of genealogies:



Ore graph, p -graph, and bipartite p -graph

paper



Properties of representations

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

p -graphs and bipartite p -graphs have many advantages:

- there are less nodes and links in p -graphs than in corresponding Ore graphs;
- p -graphs are directed, acyclic networks;
- every semi-cycle of the p -graph corresponds to a *relinking marriage*. There exist two types of relinking marriages: *blood* marriage: e.g., marriage among brother and sister, and *non-blood* marriage: e.g., two brothers marry two sisters from another family.
- p -graphs are more suitable for analyses.

Bipartite p -graphs have an additional advantage: we can distinguish between a married uncle and a remarriage of a father. This property enables us, for example, to find marriages between half-brothers and half-sisters.



Genealogies are sparse networks

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

A genealogy is *regular* if every person in it has at most two parents. Genealogies are *sparse* networks – number of links is of the same order as the number of nodes.

For a *regular Ore genealogy* $(\mathcal{V}, (\mathcal{A}, \mathcal{E}))$ we have:

$$|\mathcal{A}| \leq 2|\mathcal{V}|, \quad |\mathcal{E}| \leq \frac{1}{2}|\mathcal{V}|, \quad |\mathcal{L}| = |\mathcal{A}| + |\mathcal{E}| \leq \frac{5}{2}|\mathcal{V}|$$

p-graphs are almost trees – deviations from trees are caused by relinking marriages (\mathcal{V}_p , \mathcal{A}_p – nodes and arcs of *p*-graph, n_{mult} – # of nodes with multiple marriages):

$$|\mathcal{V}_p| = |\mathcal{V}| - |\mathcal{E}| + n_{mult}, \quad |\mathcal{V}| \geq |\mathcal{V}_p| \geq \frac{1}{2}|\mathcal{V}|, \quad |\mathcal{A}_p| \leq 2|\mathcal{V}_p|$$

and for a bipartite *p*-graph, we have

$$|\mathcal{V}| \leq |\mathcal{V}_b| \leq \frac{3}{2}|\mathcal{V}|, \quad |\mathcal{A}_b| \leq 2|\mathcal{V}| + n_{mult}$$



Number of nodes and links in Ore and p -graphs

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

data	$ \mathcal{V} $	$ \mathcal{E} $	$ \mathcal{A} $	$\frac{ \mathcal{L} }{ \mathcal{V} }$	$ \mathcal{V}_i $	n_{mult}	$ \mathcal{V}_p $	$ \mathcal{A}_p $	$\frac{ \mathcal{A}_p }{ \mathcal{V}_p }$
Drame	29606	8256	41814	1.69	13937	843	22193	21862	0.99
Hawlina	7405	2406	9908	1.66	2808	215	5214	5306	1.02
Marcus	702	215	919	1.62	292	20	507	496	0.98
Mazol	2532	856	3347	1.66	894	74	1750	1794	1.03
President	2145	978	2223	1.49	282	93	1260	1222	0.97
Royale	17774	7382	25822	1.87	4441	1431	11823	15063	1.27
Loka	47956	14154	68052	1.71	21074	1426	35228	36192	1.03
Silba	6427	2217	9627	1.84	2263	270	4480	5281	1.18
Ragusa	5999	2002	9315	1.89	2347	379	4376	5336	1.22
Tur	1269	407	1987	1.89	549	94	956	1114	1.17
Royal92	3010	1138	3724	1.62	1003	269	2141	2259	1.06
Little	25968	8778	34640	1.67	8412				1.01
Mumma	34224	11334	45565	1.66	11556				1.00
Tillson	42559	12796	54043	1.57	16967				1.00



Relinking index

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

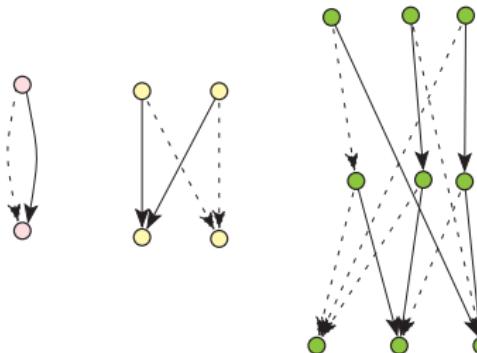
Genealogies

Pattern
searching

Triads

Motifs

Graphlets



Let n denotes number of nodes in p -graph, m number of arcs, k number of weakly connected components, and M number of maximal nodes (nodes having output degree 0, $M \geq 1$).

The *relinking index* is defined as:

$$RI = \frac{k + m - n}{k + n - 2M}$$

For a trivial graph (having only one node) we define $RI = 0$. It holds $0 \leq RI \leq 1$. $RI = 0$ iff network is a forest.



Pattern searching

Acylic
networks

V. Batagelj

Acylic
networks

Numberings

Citation
networks

Genealogies

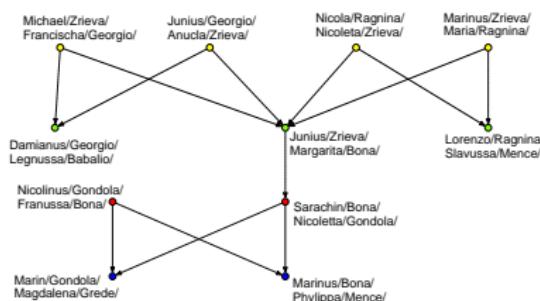
Pattern
searching

Triads

Motifs

Graphlets

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 molecules (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.



... Pattern searching

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

To speed up the search or to consider some additional properties of the pattern, a user can set some additional options:

- nodes in network should match with nodes in pattern in some nominal, ordinal or numerical property (for example, type of atom in molecule);
- values of edges must match (for example, edges representing male/female links in the case of *p-graphs*);
- the first node in the pattern can be selected only from a given subset of nodes in the network.

Networks/Fragment (First in Second)



Relinking patterns in p -graphs

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

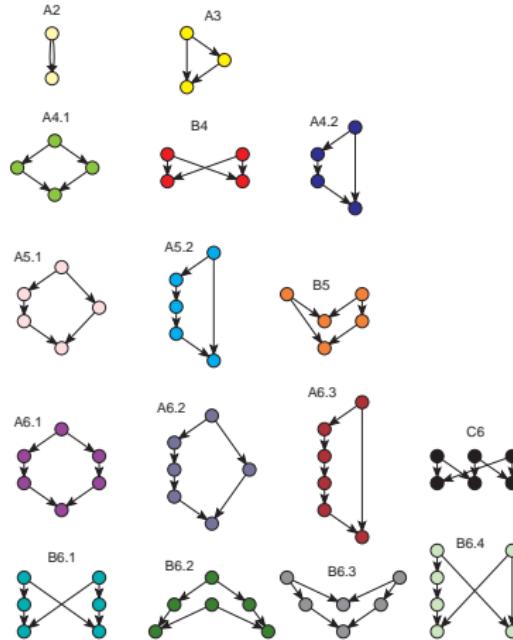
Genealogies

Pattern
searching

Triads

Motifs

Graphlets



frag16.paj

All possible relinking marriages in p -graphs with 2 to 6 nodes. Patterns are labeled as follows:

- first character – number of first nodes: A – single, B – two, C – three.
- second character: number of nodes in pattern (2, 3, 4, 5, or 6).
- last character: identifier (if the two first characters are identical).

Patterns denoted by A are exactly the blood marriages. In every pattern the number of first nodes is equal to the number of last nodes.



Frequencies normalized with number of couples in p -graph $\times 1000$

Acyclic networks

V. Batagelj

Acyclic networks

Numberings

Citation networks

Genealogies

Pattern searching

Triads

Motifs

Graphlets

pattern	Loka	Silba	Ragusa	Turcs	Royal
A2	0.07	0.00	0.00	0.00	0.00
A3	0.07	0.00	0.00	0.00	2.64
A4.1	0.85	2.26	1.50	159.71	18.45
B4	3.82	11.28	10.49	98.28	6.15
A4.2	0.00	0.00	0.00	0.00	0.00
A5.1	0.64	3.16	2.00	36.86	11.42
A5.2	0.00	0.00	0.00	0.00	0.00
B5	1.34	4.96	23.48	46.68	7.03
A6.1	1.98	12.63	1.00	169.53	11.42
A6.2	0.00	0.90	0.00	0.00	0.88
A6.3	0.00	0.00	0.00	0.00	0.00
C6	0.71	5.41	9.49	36.86	4.39
B6.1	0.00	0.45	1.00	0.00	0.00
B6.2	1.91	17.59	31.47	130.22	10.54
B6.3	3.32	13.53	40.96	113.02	11.42
B6.4	0.00	0.00	2.50	7.37	0.00
Sum	14.70	72.17	123.88	798.53	84.36

Most of the relinking marriages happened in the genealogy of Turkish nomads; the second is Ragusa while in other genealogies they are much less frequent.



Bipartite p -graphs: Marriage among half-cousins

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

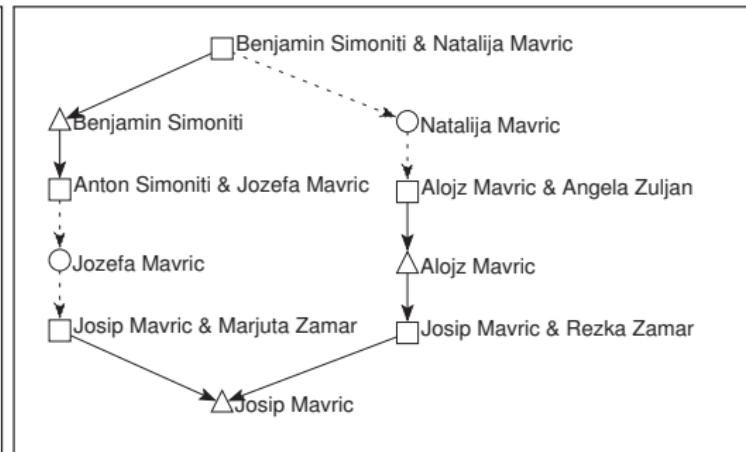
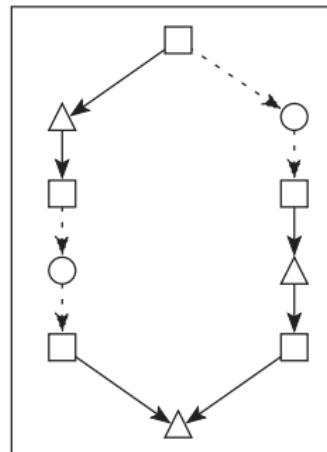
Genealogies

Pattern
searching

Triads

Motifs

Graphlets





Triads

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

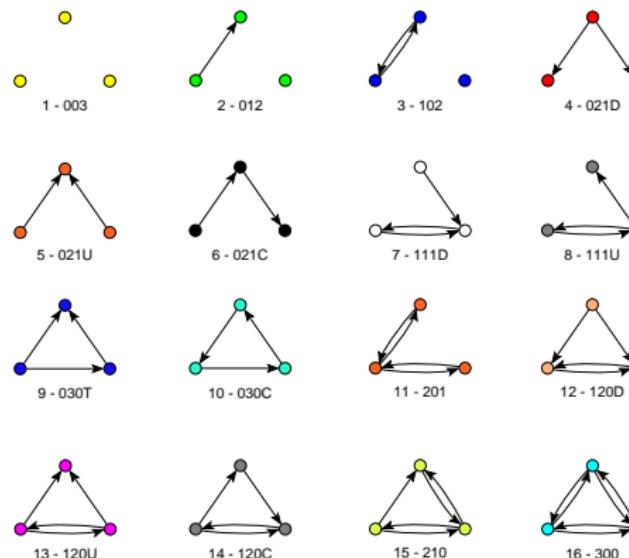
Genealogies

Pattern
searching

Triads

Motifs

Graphlets



Let $\mathcal{G} = (\mathcal{V}, \mathcal{R})$ be a simple directed graph without loops. A *triad* is a subgraph induced by a given set of three nodes. There are 16 nonisomorphic (types of) triads. They can be partitioned into three basic types:

- the *null* triad 003;
- the *dyadic* triads 012 and 102; and
- the *connected* triads:
111D, 201, 210, 300,
021D, 111U, 120D,
021U, 030T, 120U,
021C, 030C and
120C.

Network/Info/Triadic Census



Triadic spectrum

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

Triad:	BA	CL	RC	R2C	TR	HC	39+	p1	p2	p3	p4
003		✓	✓		✓	✓				✓	✓
012					✓	✓	✓			✓	✓
102	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
021D		✓	✓	✓	✓	✓					✓
021U		✓	✓	✓	✓	✓				✓	✓
021C								✓		✓	
111D											✓
111U							✓	✓			
030T		✓	✓	✓	✓	✓		✓		✓	
030C								✓	✓		✓
201											
120D		✓	✓	✓	✓	✓				✓	✓
120U		✓	✓	✓	✓	✓		✓	✓		✓
120C							✓			✓	
210						✓	✓			✓	
300	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Triad Micro-Models:

BA: Ballance (Cartwright and Harary, '56)
 CL: Clustering Model (Davis, '67)
 RC: Ranked Cluster (Davis & Leinhardt, '72)
 R2C: Ranked 2-Clusters (Johnsen, '85)
 TR: Transitivity (Davis and Leinhardt, '71)
 HC: Hierarchical Cliques (Johnsen, '85)
 39+: Model that fits D&L's 742 mats N 39-72
 p1-p4: Johnsen, 1986. Process Agreement Models.

Moody

Several properties of a graph can be expressed in terms of its **triadic spectrum** – distribution of all its triads. It also provides ingredients for p^* network models.

A direct approach to determine the triadic spectrum is of order $O(n^3)$; but in most large graphs it can be determined much faster.



Pattern counting using matrices

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

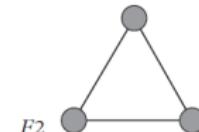
Triads

Motifs

Graphlets



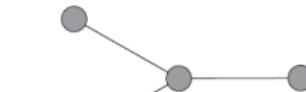
$$|F_1| = \frac{1}{2} \sum_i k_i(k_i - 1)$$



$$|F_2| = \frac{1}{6} \text{tr}(A^3)$$



$$|F_3| = \sum_{(i,j) \in E} (k_i - 1)(k_j - 1) - 3|F_2|$$



$$|F_4| = \frac{1}{6} \sum_i k_i(k_i - 1)(k_i - 2)$$

In physics they denote degrees by k .



Pattern counting using matrices

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

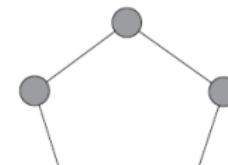
Genealogies

Pattern
searching

Triads

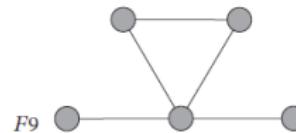
Motifs

Graphlets



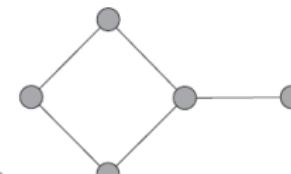
F_8

$$|F_8| = \frac{1}{10} (\text{tr}(A^5) - 30|F_2| - 10|F_6|)$$



F_9

$$|F_9| = \frac{1}{2} \sum_{k_i \geq 4} t_i(k_i - 2)(k_i - 3)$$



F_{10}

$$|F_{10}| = \frac{1}{2} \sum_i (k_i - 2) \times \sum_{i,j} \binom{(A^2)_{ij}}{2} - 2|F_7|$$



Pattern counting using matrices

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

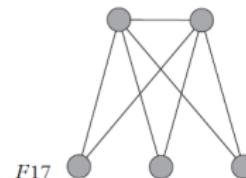
Genealogies

Pattern
searching

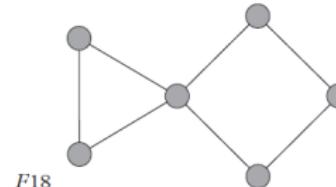
Triads

Motifs

Graphlets



$$|F_{17}| = \sum_{(i,j) \in E} \binom{(A^2)_{ij}}{3}$$



$$|F_{18}| = \sum_i t_i \cdot \sum_{i \neq j} \binom{(A^2)_{ij}}{2} - 6|F_7| - 2|F_{14}| - 6|F_{17}|$$



Motifs

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

Network motifs are sub-graphs that repeat themselves frequently in a specific network or even among various networks. Each of these sub-graphs, defined by a particular pattern of interactions between nodes, may reflect a framework in which particular functions are achieved efficiently. Indeed, motifs are of notable importance largely because they may reflect functional properties.

Milo, R, Shen-Orr, S, Itzkovitz, S, Kashtan, N, Chklovskii, D, Alon, U: Network Motifs: Simple Building Blocks of Complex Networks. Science, 298, October 2002, p. 824-827.

Wikipedia: [Motifs](#)



Motifs

Acyclic networks

V. Batagelj

Acyclic networks

Numberings

Citation networks

Genealogies

Pattern searching

Triads

Motifs

Graphlets

Network	Nodes	Edges	N_{real}	$N_{\text{rand}} \pm \text{SD}$	Z score	N_{real}	$N_{\text{rand}} \pm \text{SD}$	Z score	N_{real}	$N_{\text{rand}} \pm \text{SD}$	Z score
Gene regulation (transcription)				Feed-forward loop			Bi-fan				
<i>E. coli</i>	424	519	40	7 ± 3	10	203	47 ± 12	13			
<i>S. cerevisiae*</i>	685	1,052	70	11 ± 4	14	1812	300 ± 40	41			
Neurons				Feed-forward loop			Bi-fan			Bi-parallel	
<i>C. elegans†</i>	252	509	125	90 ± 10	3.7	127	55 ± 13	5.3	227	35 ± 10	20
Food webs				Three chain			Bi-parallel				
Little Rock	92	984	3219	3120 ± 50	2.1	7295	2220 ± 210	25			
Ythan	83	391	1182	1020 ± 20	7.2	1357	230 ± 50	23			
St. Martin	42	205	469	450 ± 10	NS	382	130 ± 20	12			
Chesapeake	31	67	80	82 ± 4	NS	26	5 ± 2	8			
Coachella	29	243	279	235 ± 12	3.6	181	80 ± 20	5			
Skipwith	25	189	184	150 ± 7	5.5	397	80 ± 25	13			
B. Brook	25	104	181	130 ± 7	7.4	267	30 ± 7	32			



Motifs

Acyclic networks

V. Batagelj

Acyclic networks

Numberings

Citation networks

Genealogies

Pattern searching

Triads

Motifs

Graphlets

Electronic circuits (forward logic chips)			X Y Z	Feed-forward loop	X Y Z W	Bi-fan	X Y Z W	Bi-parallel
s15850	10,383	14,240	424	2 ± 2	285	1040	1 ± 1	1200
s38584	20,717	34,204	413	10 ± 3	120	1739	6 ± 2	800
s38417	23,843	33,661	612	3 ± 2	400	2404	1 ± 1	2550
s9234	5,844	8,197	211	2 ± 1	140	754	1 ± 1	1050
s13207	8,651	11,831	403	2 ± 1	225	4445	1 ± 1	4950
Electronic circuits (digital fractional multipliers)			X Y Z	Three-node feedback loop	X Y Z W	Bi-fan	X → Y Z ← W	Four-node feedback loop
s208	122	189	10	1 ± 1	9	4	1 ± 1	3.8
s420	252	399	20	1 ± 1	18	10	1 ± 1	10
s838‡	512	819	40	1 ± 1	38	22	1 ± 1	20
World Wide Web			X Y Z	Feedback with two mutual dyads	X Y Z	Fully connected triad	X Y Z	Unplinked mutual dyad
nd.edu§	325,729	1.46e6	1.1e5	2e3 ± 1e2	800	6.8e6	5e4±4e2	15,000
							1.2e6	1e4 ± 2e2
								5000

R: igraph::motifs



Graphlets

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

Genealogies

Pattern
searching

Triads

Motifs

Graphlets

Graphlets are small connected non-isomorphic induced subgraphs of a large network. Graphlets differ from network motifs, since they must be induced subgraphs, whereas motifs are partial subgraphs. An induced subgraph must contain all edges between its nodes that are present in the large network, while a partial subgraph may contain only some of these edges.

Graphlets were first introduced in:

Pržulj, Nataša: Biological network comparison using graphlet degree distribution. Bioinformatics, Volume 23, Issue 2, 15 January 2007, Pages e177–e183, [PDF](#)

Wikipedia: [/Graphlets](#)

iGraph: [graphlet](#); [ORCA](#)



Graphlets with 2–5 nodes and automorphism orbits

Acyclic
networks

V. Batagelj

Acyclic
networks

Numberings

Citation
networks

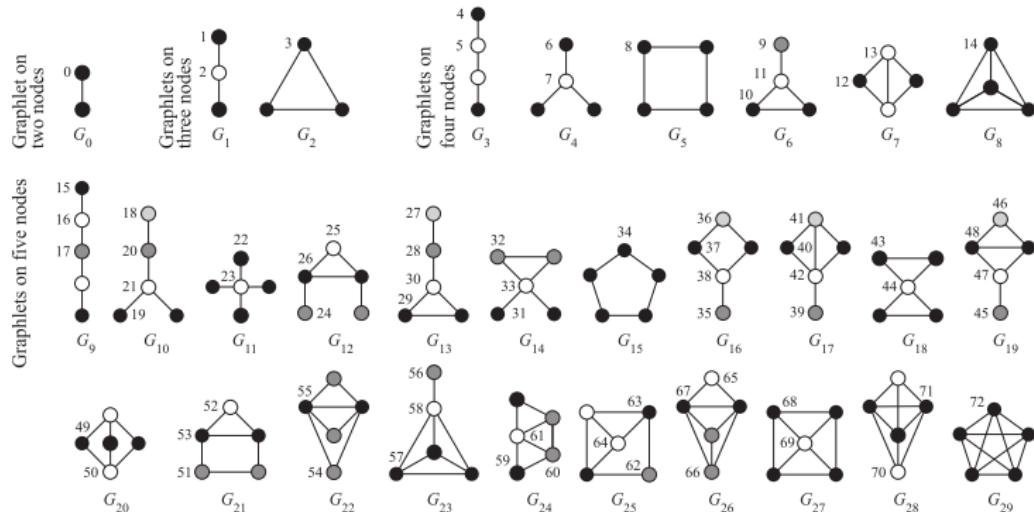
Genealogies

Pattern
searching

Triads

Motifs

Graphlets



Nodes of the same color belong to the same orbit within that graphlet.



Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Introduction to Network Analysis using **Pajek**

7. Two-mode networks and multiplication

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

PhD and MS program in Statistics
University of Ljubljana, 2022



Outline

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

Collaboration

Other derived networks

EU projects

Temporal Ns

1 Two-mode networks

2 Direct methods

3 2-mode cores

4 4-ring weights

5 Multiplication

6 Kinship relations

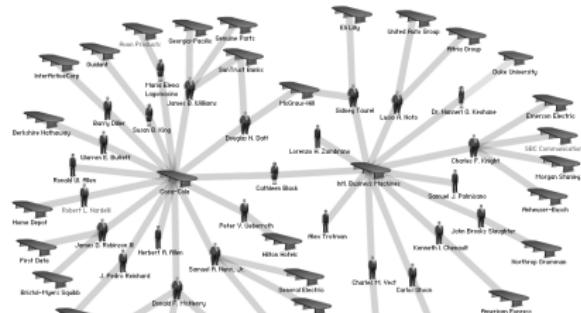
7 Projections

8 Collaboration

9 Other derived networks

10 EU projects

11 Temporal Ns



Josh On: They rule 2004

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

Current version of slides (March 23, 2022 at 00:33): [slides PDF](#)



Two-mode networks

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

In a *two-mode* network $\mathcal{N} = (\mathcal{U}, \mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ the set of nodes consists of two disjoint sets of nodes \mathcal{U} and \mathcal{V} , and all the lines from \mathcal{L} have one end-node in \mathcal{U} and the other in \mathcal{V} . Often also a *weight* $w : \mathcal{L} \rightarrow \mathbb{R} \in \mathcal{W}$ is given; if not, we assume $w(u, v) = 1$ for all $(u, v) \in \mathcal{L}$.

A two-mode network can also be described by a rectangular matrix $\mathbf{A} = [a_{uv}]_{\mathcal{U} \times \mathcal{V}}$.

$$a_{uv} = \begin{cases} w_{uv} & (u, v) \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases}$$

Examples: (persons, societies, years of membership),
(buyers/consumers, goods, quantity),
(parliamentarians, problems, positive vote),
(persons, journals, reading),
(papers, keywords, is described by), etc.



Deep South

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

Collaboration

Other derived networks

EU projects

Temporal Ns



Classical example of two-mode network are the Southern women (Davis 1941).

[Davis.paj](#). Freeman's overview.

NAMES OF PARTICIPANTS OF GROUP I	CODE NUMBERS AND DATES OF SOCIAL EVENTS REPORTED IN <i>Old City Herald</i>													
	(1) 6/27	(2) 3/2	(3) 4/12	(4) 9/26	(5) 2/25	(6) 5/19	(7) 3/15	(8) 9/16	(9) 4/6	(10) 8/10	(11) 2/23	(12) 6/7	(13) 11/21	(14) 8/3
1. Mrs. Evelyn Jefferson.....	X	X	X	X	X	X	X	X
2. Miss Laura Mandeville.....	X	X	X	X	X	X	X	X
3. Miss Theresa Anderson.....	X	X	X	X	X	X	X	X	X
4. Miss Brenda Rogers.....	X	X	X	X	X	X	X	X
5. Miss Charlotte McDowell.....	X	X	X	X
6. Miss Frances Anderson.....	X	X	X	X	X
7. Miss Eleanor Nye.....	X	X	X	X	X	X	X
8. Miss Pearl Oglethorpe.....	X	X	X	X	X	X
9. Miss Ruth DeSand.....	X	X	X	X	X	X
10. Miss Verne Sanderson.....	X	X	X	X	X
11. Miss Myra Liddell.....	X	X	X	X	X	X	X	X	X
12. Miss Katherine Rogers.....	X	X	X	X	X	X	X	X
13. Mrs. Sylvia Avondale.....	X	X	X	X	X	X	X	X
14. Mrs. Nora Fayette.....	X	X	X	X	X	X	X	X
15. Mrs. Helen Lloyd.....	X	X	X	X	X	X	X	X
16. Mrs. Dorothy Murchison.....	X	X	X	X	X	X	X	X
17. Mrs. Olivia Carlton.....	X	X
18. Mrs. Flora Price.....	X



Approaches to two-mode network analysis

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

The usual approach to analyze a two-mode network is to transform it to a one-mode network and use standard methods on it.

For direct analysis of two-mode networks we can use the *eigen-vector approach* – a two-mode variant of Kleinberg's hubs and authorities. The weight vector (\mathbf{x}, \mathbf{y}) on $\mathcal{U} \cup \mathcal{V}$ is determined by relations $\mathbf{y} = \mathbf{Ax}$ and $\mathbf{x} = \mathbf{A}^T \mathbf{y}$.

Network/2-Mode Network/Important Vertices

There are also special methods for *clustering* and *blockmodeling* in two-mode networks.

In this lecture we will present two additional direct methods: *two-mode cores* and *4-rings*.



Internet Movie Database <http://www.imdb.com/>

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

Collaboration

Other derived networks

EU projects

Temporal Ns

The Internet Movie Database

Visited by over 30 million movie lovers each month!

Welcome to the Internet Movie Database, the biggest, best, most award-winning movie site on the planet. Want to make IMDb your home page? Drag [this link](#) onto your Home button.

Honda Civic and IMDb Want You to "Pitch Your Picture" Today!

PITCH YOUR PICTURE.

You have the idea for your movie. You even have the poster. Now, [Honda Civic](#) and IMDb want you to "Pitch Your Picture." Submit your poster for your made-up movie, along with the tagline, and you may be eligible to be [entered into our "Pitch Your Picture" competition](#) (please note [game rules and restrictions](#)). We are now accepting submissions (voting will commence on the 14th). Use only your original ideas and your original images. Do not use existing screen captures, posters, or stills from other

Movie and TV News

Wed 19 October 2005:

- [Kidman Photographer Wins DNA Appeal](#)
- [Sizemore Has His Probation Reinstated](#)
- [Madonna Thanks ABBA for the Music](#)

Studio Briefing

- [Fox Obscures Box Office](#)
- [Schwarzenegger Wants To Terminate Video Game Lawsuit](#)
- [Jackson Dumps 'King Kong' Music](#)

Born Today

Wednesday, 19 October 2005:

12th Annual Graph Drawing Contest, 2005. The IMDB network is two-mode and has $1324748 = 428440 + 896308$ nodes and 3792390 arcs.





Two-mode cores

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

The subset of nodes $C \subseteq \mathcal{V}$ is a (p, q) -core in a two-mode network $\mathcal{N} = (\mathcal{V}_1, \mathcal{V}_2; \mathcal{L})$, $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ iff

- a. in the induced subnetwork $\mathcal{K} = (C_1, C_2; \mathcal{L}(C))$,
 $C_1 = C \cap \mathcal{V}_1$, $C_2 = C \cap \mathcal{V}_2$ it holds
 $\forall v \in C_1 : \deg_{\mathcal{K}}(v) \geq p$ and $\forall v \in C_2 : \deg_{\mathcal{K}}(v) \geq q$;
- b. C is the maximal subset of \mathcal{V} satisfying condition a.

Properties of two-mode cores:

- $C(0, 0) = \mathcal{V}$
- $\mathcal{K}(p, q)$ is not always connected
- $(p_1 \leq p_2) \wedge (q_1 \leq q_2) \Rightarrow C(p_1, q_1) \subseteq C(p_2, q_2)$
- $\mathcal{C} = \{C(p, q) : p, q \in \mathbb{N}\}$. If all nonempty elements of \mathcal{C} are different it is a lattice.



Algorithm for two-mode cores

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

To determine a (p, q) -core the procedure similar to the ordinary core procedure can be used:

repeat

remove from the first set all nodes of degree less than p ,

and from the second set all nodes of degree less than q

until no node was deleted

It can be implemented to run in $O(m)$ time.

Interesting (p, q) -cores? Table of cores' characteristics

$n_1 = |C_1(p, q)|$, $n_2 = |C_2(p, q)|$ and k – number of components in $\mathcal{K}(p, q)$:

- $n_1 + n_2 \leq$ selected threshold
- 'border line' in the (p, q) -table.



Table ($p, q : n_1, n_2$) for Internet Movie Database

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Network/2-Mode Network/Core/2-Mode Border

1	1590:	1590	1		16	39:	2173	678		44	14:	29	83
2	516:	788	3		17	35:	2791	995		46	13:	29	94
3	212:	1705	18		18	32:	2684	1080		49	12:	26	95
4	151:	4330	154		19	30:	2395	1063		52	11:	16	79
5	131:	4282	209		20	28:	2216	1087		56	10:	34	162
6	115:	3635	223		21	26:	1988	1087		62	9:	31	177
7	101:	3224	244		22	24:	1854	1153		66	8:	29	198
8	88:	2860	263		24	23:	34	39		72	7:	22	203
9	77:	3467	393		27	22:	31	38		96	6:	7	114
10	69:	3150	428		29	20:	35	52		119	5:	6	137
11	63:	2442	382		32	19:	34	57		141	4:	8	258
12	56:	2479	454		35	18:	33	61		186	3:	3	186
13	50:	3330	716		36	17:	33	65		247	2:	2	247
14	46:	2460	596		39	16:	29	70		1334	1:	1	1334
15	42:	2663	739		42	15:	28	76					



(247,2)-core and (27,22)-core

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

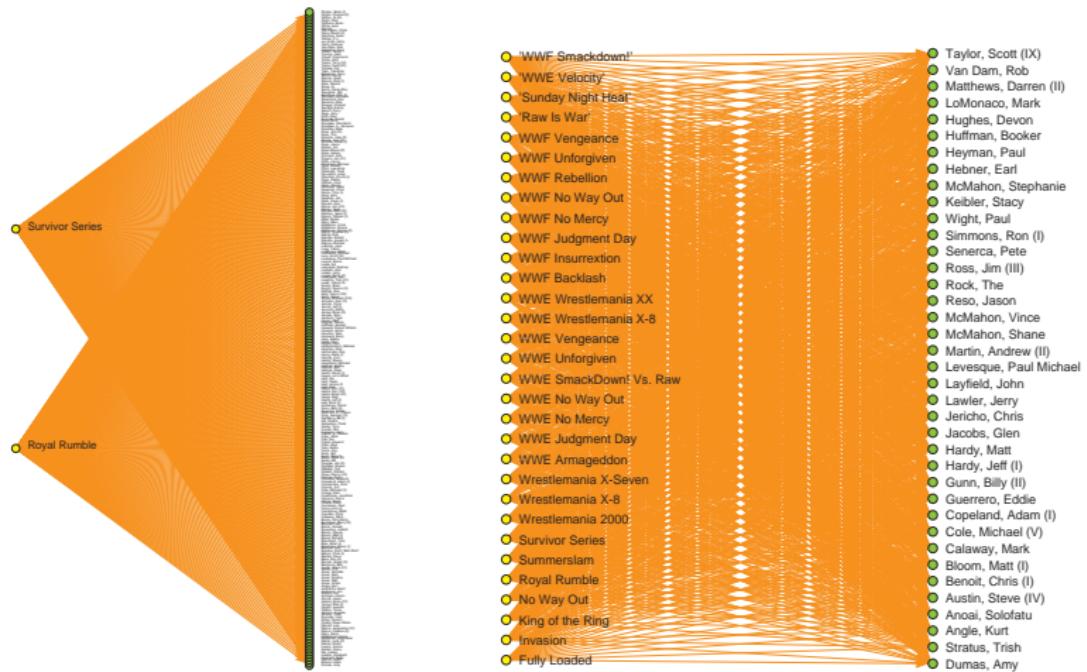
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns





(2,516)-Hard core

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

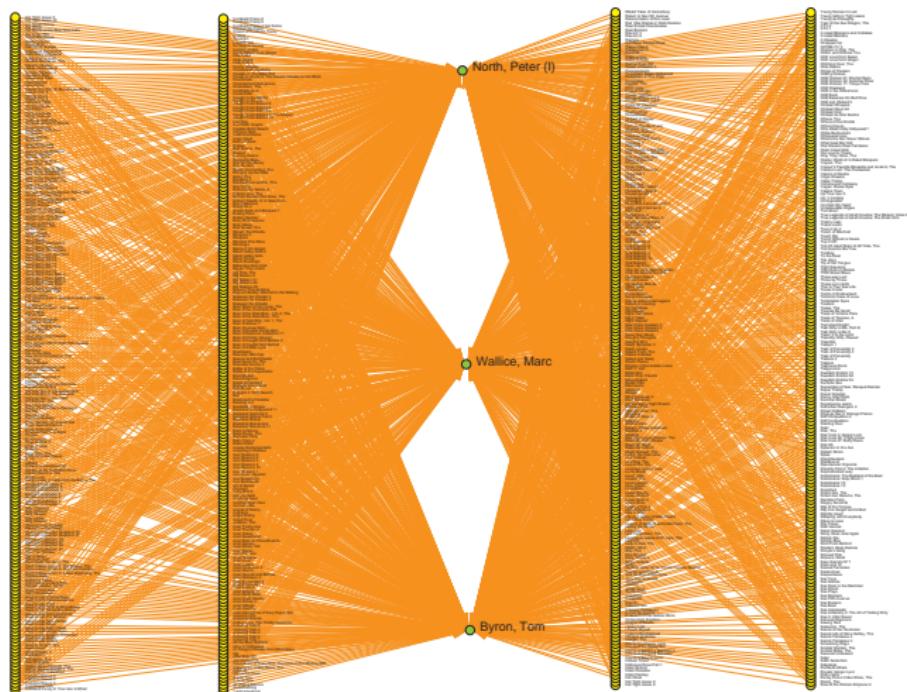
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns





IMDB cores / Pajek commands

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

```
Options/Read-Write/Read-Save vertices labels [Off]
Read/Network [IMDB.net] 1:40
Info/Memory
Network/2-Mode Network/Core/2-Mode Review
Network/2-Mode Network/Core/2-Mode [27 22]
Info/Partition
Operations/Network+Partition/Extract Subnetwork [Yes 1]
Network/2-Mode Network/Partition into 2 Modes
Network/Create New Network/Transform/Add/Vertex Labels/
    from File(s) [IMDB.nam]
Draw/Network+First Partition
Layers/in y direction
Options/Transform/Rotate 2D [90]
```



k-rings

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

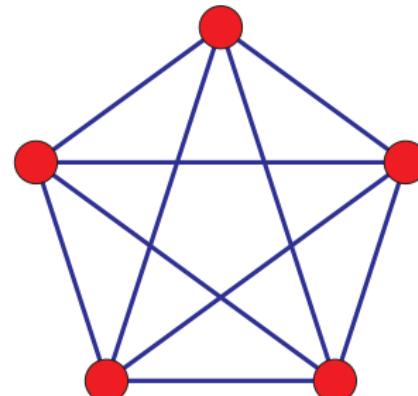
Other derived
networks

EU projects

Temporal Ns

A *k*-ring is a simple closed chain of length k . Using *k*-rings we can define a weight of edges as

$$w_k(e) = \# \text{ of different } k\text{-rings containing the edge } e \in \mathcal{E}$$



Complete graph K_5

Since for each edge e of a complete graph K_r , $r \geq k \geq 3$ we have $w_k(e) = (r-2)!/(r-k)!$ the edges belonging to cliques have large weights. Therefore these weights can be used to identify the dense parts of a network.

The *k*-rings can be efficiently determined only for small values of $k = 3, 4, 5$.

On the *k*-rings we can also base the notion of short cycle connectivity which provides us with another decomposition of networks. [paper](#)



4-rings and analysis of two-mode networks

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

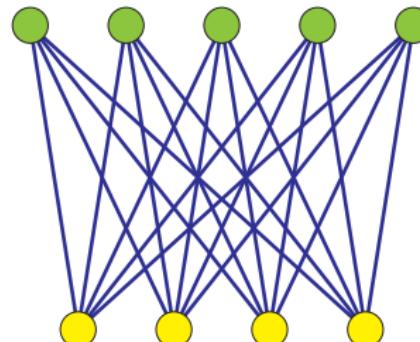
Other derived
networks

EU projects

Temporal Ns

In two-mode network there are no 3-rings. The densest substructures are complete bipartite subgraphs $K_{p,q}$. They contain many 4-rings.

There are



$$\binom{p}{2} \binom{q}{2} = \frac{1}{4} p(p-1)q(q-1)$$

4-rings in $K_{p,q}$; and each of its edges e has weight

$$w_4(e) = (p-1)(q-1)$$

Network/Create New Network/with Ring Counts.../4-Rings/Undirected



Directed 4-rings

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

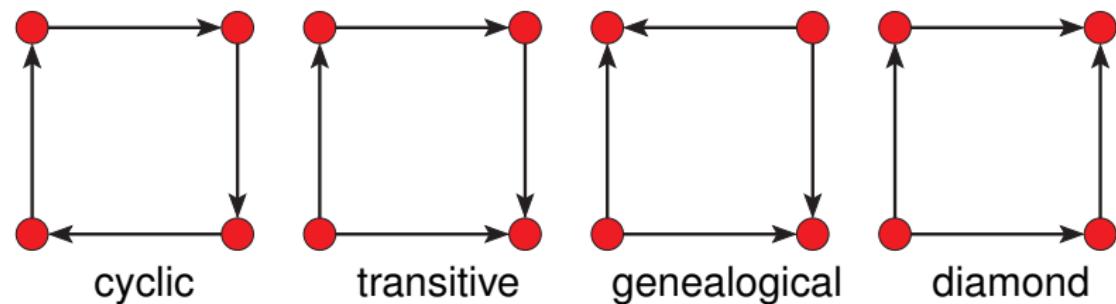
Collaboration

Other derived
networks

EU projects

Temporal Ns

There are 4 types of directed 4-rings:



In the case of transitive rings **Pajek** provides a special weight counting on how many transitive rings the arc is a *shortcut*.

Network/Create New Network/with Ring

Counts/4-Rings/Directed



Simple line islands in IMDB for w_4

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

We obtained 12465 simple line islands on 56086 nodes.
Here is their size distribution.

	Size	Freq		Size	Freq		Size	Freq		Size	Freq
Direct methods	2	5512		20	19		38	4		59	2
2-mode cores	3	1978		21	18		39	3		61	1
4-ring weights	4	1639		22	15		40	2		64	1
Multiplication	5	968		23	9		42	2		67	1
Kinship relations	6	666		24	13		43	3		70	1
Projections	7	394		25	12		45	3		73	1
Collaboration	8	257		26	6		46	4		76	1
Other derived networks	9	209		27	6		47	5		82	1
EU projects	10	148		28	5		48	1		86	1
Temporal Ns	11	118		29	6		49	2		106	1
	12	87		30	3		50	2		122	1
	13	55		31	6		51	1		135	1
	14	62		32	5		52	2		144	1
	15	46		33	3		53	1		163	1
	16	39		34	1		54	2		269	1
	17	27		35	5		55	1		301	1
	18	28		36	4		57	1		332	2
	19	29		37	7		58	1		673	1



Example: Islands for w_4 Charlie Brown and Adult

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

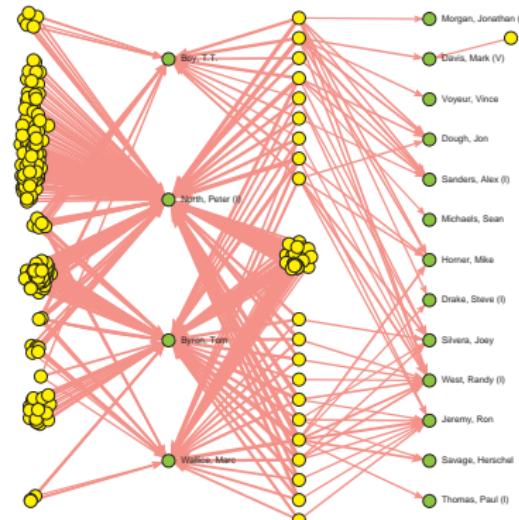
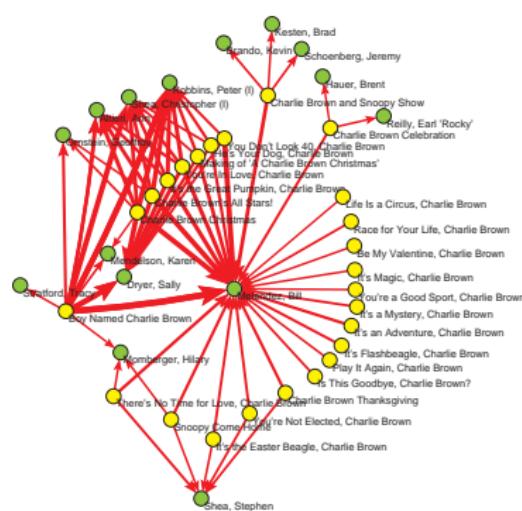
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns





Example: Islands for w_4

Mark Twain and Abid

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

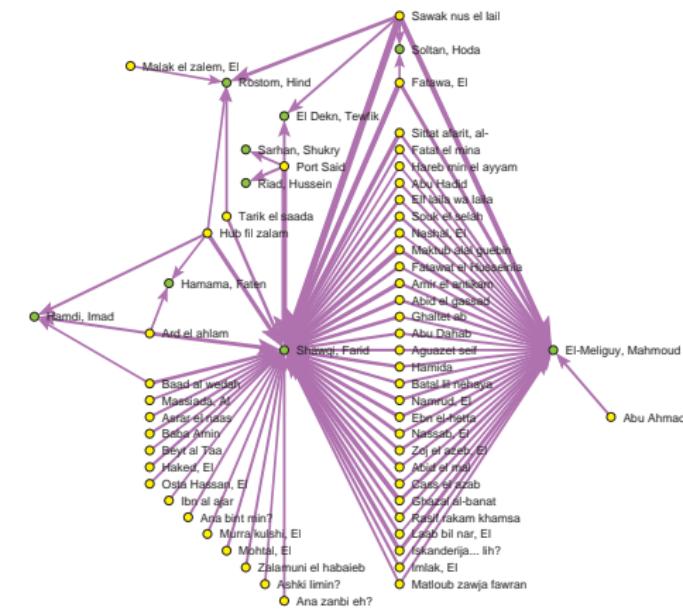
Projections

Collaboration

Other derived networks

EU projects

Temporal Ns





Example: Island for w_4 Polizeiruf 110 and Starkes Team

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

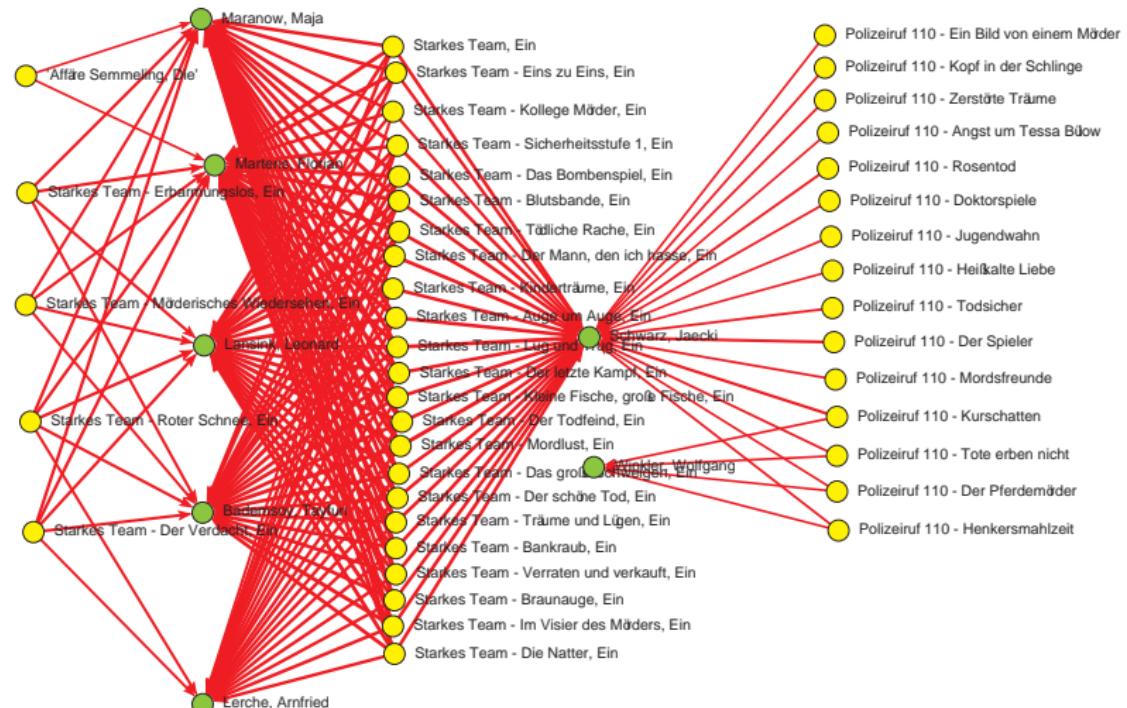
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns





5-rings

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

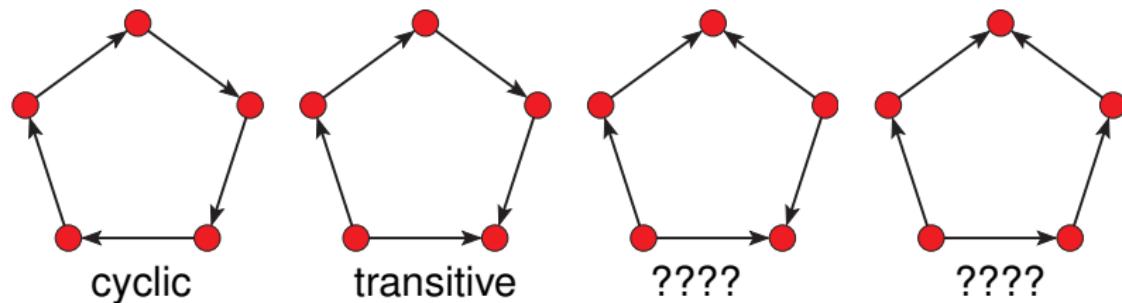
Collaboration

Other derived
networks

EU projects

Temporal Ns

In the future we intend to implement in **Pajek** also weights w_5 . Again there are only 4 types of directed 5-rings.





Two mode networks from data tables

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

RuthDELmain.csv

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Ident	Num	File	ORGANISATION ORG	Org Contact Name	Street	ZIP	Project	City	Country	coun	EU	Region	
1	1	1480 613.html	3D PLUS SA	3D F3D I LIGNIER, Olivier	641 Ru	78530	IST-2001-3440	Buc	FRANCE	20	2	ÎLE DE FR	
2	2	1481 613.html	3D PLUS SA	3D PLUS LIGNIER, Olivier	641 Ru	78530	IST-2001-3440	Buc	FRANCE	20	2	ÎLE DE FR	
3	3	4001 924.html	3D VISION	3D V3D MARIAT, Jacques	Savoie	73375		502909	Le Bois	FRANCE	20	2 CENTRE-I	
4	4	1648 160.html	3D Web Technologies	3D WEB DENNISON, Andrew	M31 4XL	BMH4989519		Carrir	UNITED KINGDOM	60	2	NORTH W.	
5	5	1406 442.html	3E	PALMERS, Geert	Eredier	1000	NNE5/51/1999	Bruxelles	BELGIQUE	8	2	REG.BRU	
6	6	1007 884.html	4M2C PATRIC SALOMON	4M2C P/N/A	CRANB	12157		507255	Berlin	DEUTSCH	15	2 BERLIN E	
7	7	7914 991.html	5T S.c.r.l.	5T S.C.R.N/A	C.so B	10126	Read2/506716	Torino	ITALIA	26	2	NORD ÖV	
8	8	6880 588.html	A & C 2000 S.R.L	A & C SAN CARLUCCI, Renzo	VIALE	148	IST-2001-3454	Roma	ITALIA	26	2	LAZIO R	
9	9	6881 588.html	A & C 2000 S.R.L	A & C 2000 S.R.L	RENZ Viale C	148	IST-2001-3454	Roma	ITALIA	26	2	LAZIO R	
10	10	1647 176.html	A. BENETTI MACCHIA	A. BENETTI MACCHIA	Federico BENETTI Via Pro	54033	BRST985466	Carra	ITALIA	26	2	CENTRO	
11	11	6605 984.html	A. Mickiewicz Universitatis	A. MInst PATKOWSKI, Ad UU	H. 161-712		502235	Pozn	POLSKA	45	2		
12	12	6571 135.html	A.BRITO - INDUSTRIAL	A.BRITO VIEIRA DE BRIT	(5109.E4350-115)	BRST985263		Porto	PORTUGA	46	2	CONTINENT	
13	13	1813 409.html	A.L. DIGITAL LIMITED	A.L. A.L. LAURIE, Ben	VOYSEIW4	4GB	IST-2000-2633	Chisinau	UNITED KINGDOM	60	2	SOUTH E.	
14	14	1814 409.html	A.L. Digital Limited	A.L. DIG LAURIE, Ben	VOYSEIW4	4GB	IST-2000-2633	Chisinau	UNITED KINGDOM	60	2	SOUTH E.	
15	15	1885 960.html	A.P. MOLLER-MAER	A.P. TEC DRAGSTED, Jørn	Esplan	1098		506876	København	DANMARK	14	2 København	
16	16	6731 537.html	A.S.M. S.A.	A.S.M. S. MOYA GARCIA,	Carretera	43206	IST-2000-3008	Reus	ESPAÑA	19	2	ESTE CA	
17	17	8150 232.html	AABO AKADEMI UNI	AAB CO NYBACKA-WILL	WILL 14-18B	20500	ERK5-CT-1995	Turku	SUOMI/FIN	53	2	MANNER-	
18	18	8152 662.html	AABO AKADEMI UNI	AAB DEF BJORKSTRAND	3,Tykis	20521	EVK1-CT-2002	Turku	SUOMI/FIN	53	2		
19	19	8148 959.html	AABO AKADEMI UNI	AAB Def HUPA, Mikko	Domkyro	20500		502679	Turku	SUOMI/FIN	53	2	MANNER-
20	20	8151 233.html	AABO AKADEMI UNI	AAB DEF NYBACKA-WILL	WILL Lemmijoki	20500	ERK6-CT-1995	Turku	SUOMI/FIN	53	2	MANNER-	
21	21	125 116.html	AACHEN UNIVERSITÄT	AAC GIE E. NEUSSL	Intzest	52072	BRPR980663	Aach	DEUTSCH	15	2	NORDRH	
22	22	123 104.html	AACHEN UNIVERSITÄT	AAC GIE MEISER, Lukas	Intzest	52072	BRPR980695	Aach	DEUTSCH	15	2	NORDRH	
23	23	155 364.html	AACHEN UNIVERSITÄT	AAC INS RAJAHUT, Burkha	Elif	52062	GIRD-CT-200	Aach	DEUTSCH	15	2	NORDRH	

A *data table* \mathcal{T} is a set of *records* $\mathcal{T} = \{T_k : k \in \mathcal{K}\}$, where \mathcal{K} is the set of *keys*. A record has the form $T_k = (k, q_1(k), q_2(k), \dots, q_r(k))$ where $q_i(k)$ is the value of the *property* (attribute) q_i for the key k .



... Two mode networks from data tables

Suppose that the property \mathbf{q} has the range $2^{\mathcal{Q}}$. For example:

$\text{Authors}(\text{SNA}) = \{ \text{S. Wasserman}, \text{K. Faust} \}$,

$\text{PubYear}(\text{SNA}) = \{ 1994 \}$,

$\text{Keywords}(\text{SNA}) = \{ \text{network, centrality, matrix, ...} \}, \dots$

If \mathcal{Q} is finite (it can always be transformed into such set by partitioning the set \mathcal{Q} and recoding the values) we can assign to the property \mathbf{q} a two-mode network $\mathcal{K} \times \mathbf{q} = (\mathcal{K}, \mathcal{Q}, \mathcal{E}, w)$ where $(k, v) \in \mathcal{E}$ iff $v \in q(k)$, and $w(k, v) = 1$.

	...	Bata gelj	Faust	de Nooy	Kej žar	Kore njak	Mrvar	Wasse rman	Zaver šnik	...
...	GenCores		1						1	
	Islands		1						1	
	ESNA2			1				1		
	IFCS09		1		1	1				
	SNA			1					1	
	...									

Single-valued properties can be represented by a partition.

We can always transform the partition into corresponding network.



Record from Web of Science

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

PT J
AU Dipple, H
Evans, B
TI The Leicestershire Huntington's disease support group: a social network analysis
SO HEALTH & SOCIAL CARE IN THE COMMUNITY
LA English
DT Article
C1 Rehabil Serv, Troon Way Business Ctr, Leicester LE4 9HA, Leics, England.
RP Dipple, H, Rehabil Serv, Troon Way Business Ctr, Sandringham Suite, Humberstone Lane, Leicester LE4 9HA, Leics, England.
CR BORGATTI SP, 1992, UCINET 4 VERSION 1 0
FOLSTEIN S, 1989, HUNTINGTONS DIS DISO
SCOTT J, 1991, SOCIAL NETWORK ANAL
NR 3
TC 3
PU BLACKWELL SCIENCE LTD
PI OXFORD
PA P O BOX 88, OSNEY MEAD, OXFORD OX2 0NE, OXON, ENGLAND
SN 0966-0410
J9 HEALTH SOC CARE COMMUNITY
JI Health Soc. Care Community
PD JUL
PY 1998
VL 6
IS 4
BP 286
EP 289
PG 4
SC Public, Environmental & Occupational Health; Social Work
GA 105UP
UT ISI:000075092200008
ER

WoS2Pajek



Records from BiBTeX

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

```
@Article{int:Mizunol,  
    author = "S. Mizuno",  
    title = "An  $O(n^3L)$  algorithm using a sequence fo  
    linear complementarity problems",  
    journal = "Journal of the Operations Research Society of  
    volume = "33",  
    year = "1990",  
    pages = "66--75",  
}  
  
@InCollection{int:Vorstl,  
    author = "{J. G. G. van de} Vorst",  
    title = "An attempt to use parallel computing in large  
    optimisation",  
    booktitle = "Logistics, Where Ends Have to Meet": Proceeding  
    the Shell Conference on Logistics in Apeldoorn,  
    Netherlands, November 1988",  
    editor = "{C. F. H. van} Rijn",  
    year = "1989",  
    pages = "112--119",  
    publisher = "Pergamon Press",  
    address = "Oxford, United Kingdom",  
}
```

Bib2Pajek.py



Two mode networks from data tables

For data from the **Web of Science** (Knowledge) we can obtain the corresponding networks using the program **WoS2Pajek**:

- citation network **Ci**: works \times works;
- authorship network **WA**: works \times authors, for works without complete description only the first author is known;
- keywords network **WK**: works \times keywords, only for works with complete description;
- journals network **WJ**: works \times journals;
- partition of works by the publication year;
- partition of works – complete description (1) / ISI name only (0);

Similar programs exist also for other bibliographic sources/formats: Scopus, BibTeX, Zentralblatt Math, Google Scholar, DBLP, IMDB, etc.



Linked / multi-modal networks

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Linked or multi-modal networks are collections of networks over at least two sets of nodes (modes) and consist of some one-mode networks and some two-mode networks linking different modes. For example: modes are Persons and Organizations. Two one-mode networks describe collaboration among Persons and among Organizations. The linking two-mode network describes membership of Persons to different Organizations.

An important approach in analysis of linked networks is the use of derived networks obtained by network multiplication.

- Krackhardt, D., Carley, K.M. 1998. A PCANS Model of Structure in Organization. In Proceedings of the 1998 International Symposium on Command and Control Research and Technology Evidence Based Research: 113-119, Vienna, VA. [MetaMatrix](#), [paper](#)
- Kathleen M. Carley (2003). Dynamic Network Analysis. in the Summary of the NRC workshop on Social Network Modeling and Analysis, Ron Breiger and Kathleen M. Carley (Eds.), National Research Council. [preprint](#)



MetaMatrix

Carley and Diesner

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

Collaboration

Other derived networks

EU projects

Temporal Ns

Meta-Matrix Entities	Agent	Knowledge	Resources	Tasks/ Event	Organizations	Location
Agent	Social network	Knowledge network	Capabilities network	Assignment network	Membership network	Agent location network
Knowledge		Information network	Training network	Knowledge requirement network	Organizational knowledge network	Knowledge location network
Resources			Resource network	Resource requirement Network	Organizational Capability network	Resource location network
Tasks/ Events				Precedence network	Organizational assignment network	Task/Event location network
Organizations					Inter-organizational network	Organizational location network
Location						Proximity network



Multiplication of networks

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

To a simple (no parallel arcs) two-mode *network*

$\mathcal{N} = (\mathcal{I}, \mathcal{J}, \mathcal{A}, w)$; where \mathcal{I} and \mathcal{J} are sets of *nodes*, \mathcal{A} is a set of *arcs* linking \mathcal{I} and \mathcal{J} , and $w : \mathcal{A} \rightarrow \mathbb{R}$ (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 $\mathcal{N}_A = (\mathcal{I}, \mathcal{K}, \mathcal{A}_A, w_A)$ and $\mathcal{N}_B = (\mathcal{K}, \mathcal{J}, \mathcal{A}_B, w_B)$ with corresponding matrices $\mathbf{A}_{\mathcal{I} \times \mathcal{K}}$ and $\mathbf{B}_{\mathcal{K} \times \mathcal{J}}$ we call a *product of networks* \mathcal{N}_A and \mathcal{N}_B a network

$\mathcal{N}_C = (\mathcal{I}, \mathcal{J}, \mathcal{A}_C, w_C)$, 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

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

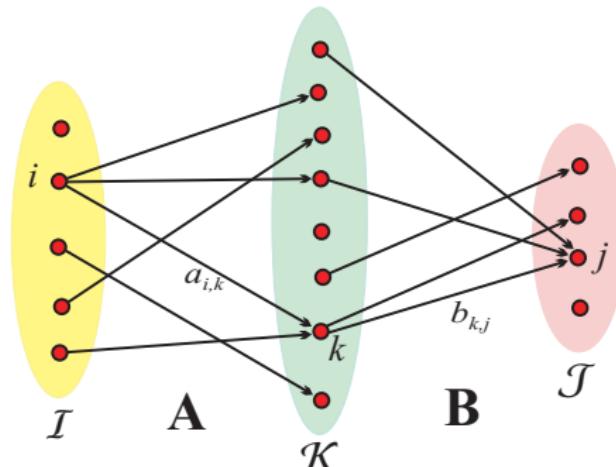
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns



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

If all weights in networks \mathcal{N}_A and \mathcal{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

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

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 K do
    for (i,j) in N_A^-(k) × N_B(k) do
        if ∃ c_{i,j} then c_{i,j} := c_{i,j} + a_{i,k} · b_{k,j}
        else new c_{i,j} := a_{i,k} · 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

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

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 \mathcal{N}_A and \mathcal{N}_B has small maximal degree on \mathcal{K} then also the resulting product network \mathcal{N}_C is sparse.

If for the sparse networks \mathcal{N}_A and \mathcal{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 \mathcal{N}_C is sparse.



Kinship relations

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Anthropologists typically use a basic vocabulary of kin types to represent genealogical relationships. One common version of the vocabulary for basic relationships:

Kin Type	English Type
P	Parent
F	Father
M	Mother
C	Child
D	Daughter
S	Son
G	Sibling
Z	Sister
B	Brother
E	Spouse
H	Husband
W	Wife

The genealogies are usually described in **GEDCOM** format.

Examples **family**, **Bouchards**. **Paper**



Calculating kinship relations

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Pajek generates three relations when reading genealogy as Ore graph:

F: $_ \text{ is a father of } _$

M: $_ \text{ is a mother of } _$

E: $_ \text{ is a spouse of } _$

Additionally we must generate two binary diagonal matrices, to distinguish between male and female:

L: $_ \text{ is a male } _ / 1\text{-male, } 0\text{-female}$

J: $_ \text{ is a female } _ / 1\text{-female, } 0\text{-male}$

$$\mathbf{F} \cap \mathbf{M} = \emptyset, \quad \mathbf{L} \cup \mathbf{J} \subseteq \mathbf{I}, \quad \mathbf{L} \cap \mathbf{J} = \emptyset$$



Derived kinship relations

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Other basic relations can be obtained using macros based on identities:

<u><i>is a parent of</i></u>	$P = F \cup M$
<u><i>is a child of</i></u>	$C = P^T$
<u><i>is a son of</i></u>	$S = L * C$
<u><i>is a daughter of</i></u>	$D = J * C$
<u><i>is a husband of</i></u>	$H = L * E$
<u><i>is a wife of</i></u>	$W = J * E$
<u><i>is a sibling of</i></u>	$G = ((F^T * F) \cap (M^T * M)) \setminus I$
<u><i>is a brother of</i></u>	$B = L * G$
<u><i>is a sister of</i></u>	$Z = J * G$
<u><i>is an uncle of</i></u>	$U = B * P$
<u><i>is an aunt of</i></u>	$A = Z * P$
<u><i>is a semi-sibling of</i></u>	$G_e = (P^T * P) \setminus I$

and using them other relations can be determined

<u><i>is a grand mother of</i></u>	$M_2 = M * P$
<u><i>is a niece of</i></u>	$Ni = D * G$



Relative sizes of kinship relations in genealogies

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Kin Type	Turks	Ragusa	Loka	Silba	Royal
P-Parent	1.000	1.000	1.000	1.000	1.000
F-Father	0.514	0.532	0.504	0.519	0.540
M-Mother	0.486	0.468	0.496	0.481	0.460
C-Child	1.000	1.000	1.000	1.000	1.000
D-Daughter	0.431	0.384	0.480	0.469	0.427
S-Son	0.569	0.616	0.520	0.531	0.573
G-Sibling	1.250	0.943	1.019	0.811	0.767
Z-Sister	1.135	0.746	0.983	0.760	0.707
B-Brother	1.366	1.140	1.055	0.861	0.828
E-Spouse	0.205	0.215	0.208	0.230	0.306
H-Husband	0.205	0.215	0.208	0.230	0.306
W-Wife	0.205	0.215	0.208	0.230	0.306
U-Uncle	1.920	1.789	1.200	1.181	0.927
A-Aunt	1.750	1.143	1.190	1.097	0.798
Ge-Semi-sibling	1.473	1.155	1.128	0.932	0.905
n	1269	5999	47956	6427	3010
mE = Spouse	407	2002	14154	2217	1138
mA = Parent	1987	9315	68052	9627	3724



Two-mode network analysis by conversion to one-mode network

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Often we transform a two-mode network $\mathcal{N} = (\mathcal{U}, \mathcal{V}, \mathcal{E}, w)$ into an ordinary (one-mode) network $\mathcal{N}_1 = (\mathcal{U}, \mathcal{E}_1, w_1)$ or/and $\mathcal{N}_2 = (\mathcal{V}, \mathcal{E}_2, w_2)$, where \mathcal{E}_1 and w_1 are determined by the matrix $\mathbf{W}^{(1)} = \mathbf{W}\mathbf{W}^T$, $w_{uv}^{(1)} = \sum_{z \in \mathcal{V}} w_{uz} \cdot w_{zv}^T$. Evidently $w_{uv}^{(1)} = w_{vu}^{(1)}$. There is an edge $(u : v) \in \mathcal{E}_1$ in \mathcal{N}_1 iff $N(u) \cap N(v) \neq \emptyset$. Its weight is $w_1(u, v) = w_{uv}^{(1)}$.

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

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

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



Normalizations

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

Collaboration

Other derived networks

EU projects

Temporal Ns

The *normalization* approach was developed for quick inspection of (1-mode) networks obtained from two-mode networks – a kind of network based data-mining.

In networks obtained from large two-mode networks there are often huge differences in weights. Therefore it is not possible to compare the vertices according to the raw data. 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 table. They can be used also on other networks.

In the case of networks without loops we define the diagonal weights for undirected networks as the sum of out-diagonal elements in the row (or column) $w_{vv} = \sum_u w_{vu}$ and for directed networks as some mean value of the row and column sum, for example $w_{vv} = \frac{1}{2}(\sum_u w_{vu} + \sum_u w_{uv})$. Usually we assume that the network does not contain any isolated node.



... Normalizations

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

$$\begin{array}{ll}
 \text{Geo}_{uv} = \frac{w_{uv}}{\sqrt{w_{uu} w_{vv}}} & \text{GeoDeg}_{uv} = \frac{w_{uv}}{\sqrt{\deg_u \deg_v}} \\
 \text{Input}_{uv} = \frac{w_{uv}}{w_{vv}} & \text{Output}_{uv} = \frac{w_{uv}}{w_{uu}} \\
 \text{Min}_{uv} = \frac{w_{uv}}{\min(w_{uu}, w_{vv})} & \text{Max}_{uv} = \frac{w_{uv}}{\max(w_{uu}, w_{vv})} \\
 \text{MinDir}_{uv} = \begin{cases} \frac{w_{uv}}{w_{uu}} & w_{uu} \leq w_{vv} \\ 0 & \text{otherwise} \end{cases} & \text{MaxDir}_{uv} = \begin{cases} \frac{w_{uv}}{w_{vv}} & w_{uu} \leq w_{vv} \\ 0 & \text{otherwise} \end{cases}
 \end{array}$$

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

Network / 2-Mode Network / 2-Mode to 1-Mode / Normalize 1-Mode

Reuters Terror News: **GeoDeg, MaxDir, MinDir.**

Slovenian journals and magazins.



MinDir of Slovenian journals 2000

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

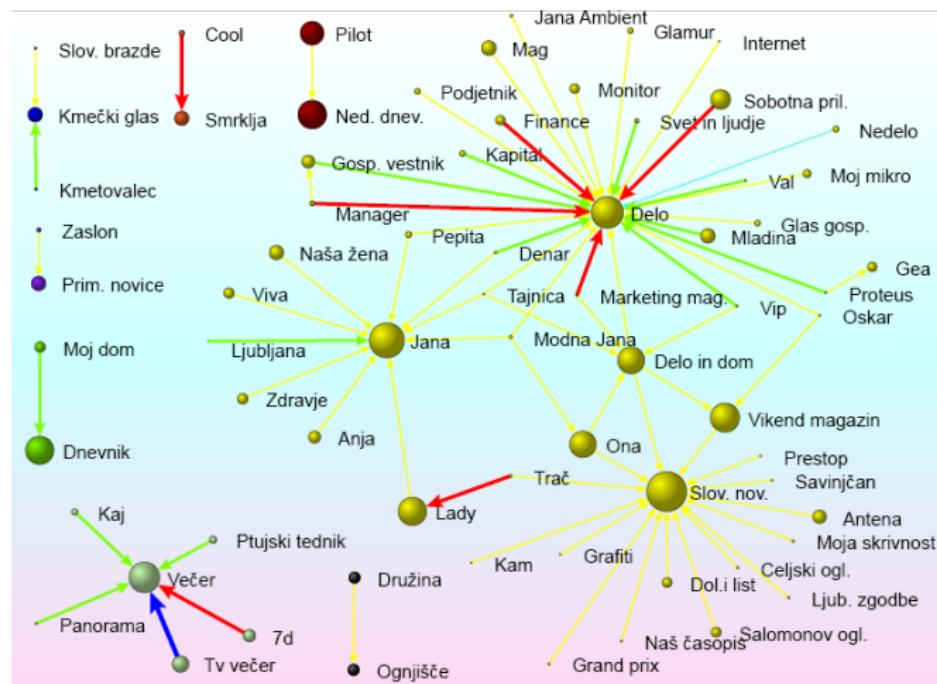
Projections

Collaboration

Other derived networks

EU projects

Temporal Ns



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



GeoDeg normalization of Reuters terror news network

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

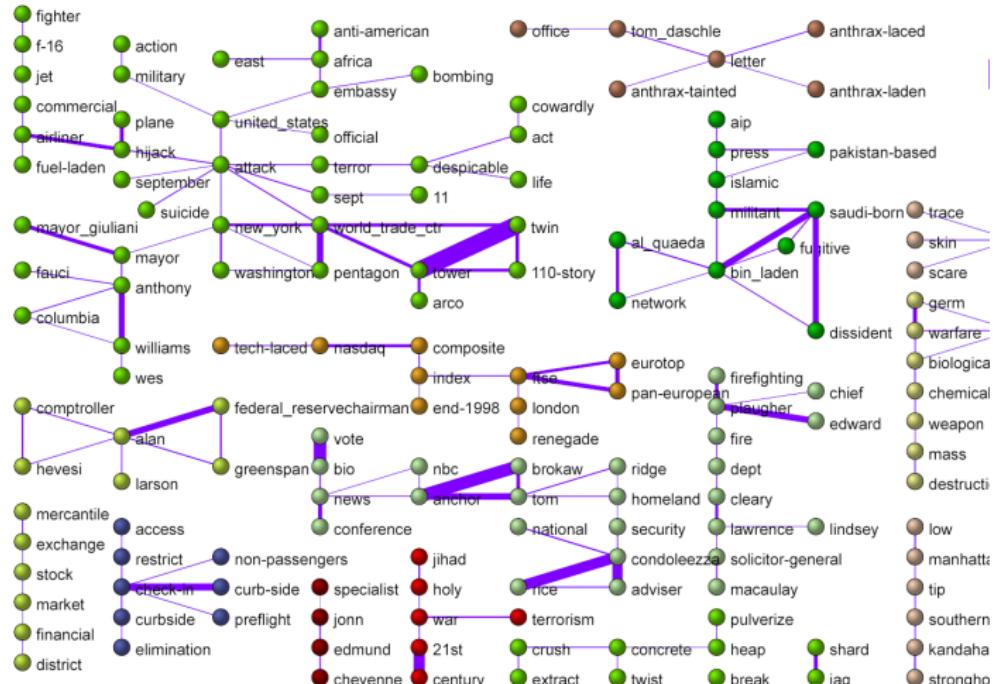
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns





Authorship network

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Let **WA** be the works \times authors two mode authorship network; $wa_{pi} \in \{0, 1\}$ is describing the authorship of author i of work p .

$$\forall p \in W : \sum_{i \in A} wa_{pi} = \text{outdeg}_{WA}(p) = \# \text{ authors of work } p$$

Let **N** be its normalized version

$$\forall p \in W : \sum_{i \in A} n_{pi} \in \{0, 1\}$$

obtained from **WA** by $n_{pi} = wa_{pi} / \max(1, \text{outdeg}_{WA}(p))$, or by some other rule determining the author's contribution.



Some transformations of networks

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Binarization $b(\mathcal{N})$ is a network obtained from the \mathcal{N} in which all weights are set to 1.

Transposition \mathcal{N}^T or $t(\mathcal{N})$ is a network obtained from \mathcal{N} in which to all arcs their direction is reversed. $\mathbf{AW} = \mathbf{WA}^T$, $\mathbf{KW} = \mathbf{WK}^T$, ...

(Out) normalization $n(\mathcal{N})$ is a network obtained from \mathcal{N} in which the weight of each arc a is divided by the sum of weights of all arcs having the same initial vertex as the arc a . For binary networks

$$n(\mathbf{A}) = \text{diag}\left(\frac{1}{\max(1, \text{outdeg}_{WA}(i))}\right)_{i \in \mathcal{I}} * \mathbf{A}$$

$$\mathbf{N} = n(\mathbf{WA}), \mathbf{WA} = b(\mathbf{N})$$



First co-authorship network

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

$$\mathbf{Co} = \mathbf{AW} * \mathbf{WA}$$

$$co_{ij} = \sum_{p \in W} wa_{pi} wa_{pj} = \sum_{p \in N^-(i) \cap N^-(j)} 1$$

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

$$\cos(i, j) = \frac{co_{ij}}{\sqrt{co_{ii}co_{jj}}}, \quad \text{for } co_{ij} > 0$$



Cores of orders 20–47 in $\text{Co}(\text{SN5})$

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

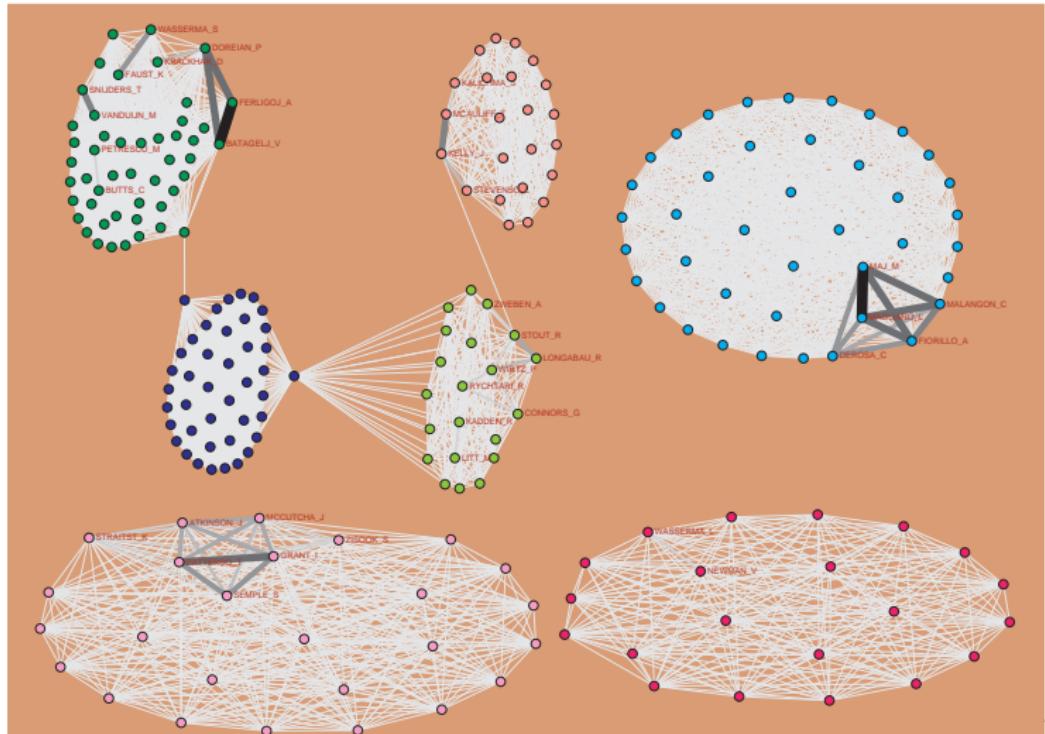
Collaboration

Other derived networks

EU projects

Temporal Ns

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



V. Batagelj

Two-mode networks



Papers by number of authors

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

Collaboration

Other derived networks

EU projects

Temporal Ns

Problem: The **Co** network is composed of complete graphs on the set of work's authors. Works with many authors produce large complete subgraphs and are over-represented, thus blurring the collaboration structure.

	outdeg	frequency		outdeg	frequency	paper
	1	2637		12	8	
	2	2143		13	4	
	3	1333		14	3	
	4	713		15	2	
	5	396		21	1	Pierce et al. (2007)
	6	206		22	1	Allen et al. (1998)
	7	114		23	1	Kelly et al. (1997)
	8	65		26	1	Semple et al. (1993)
	9	43		41	1	Magliano et al. (2006)
	10	24		42	1	Doll et al. (1992)
	11	10		48	1	Snijders et al. (2007)



Snijders et al. (2007)

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Snijders et al.(2007): Snijders, T.A.B., Robinson, T., Atkinson, A.C., Riani, M., Gormley, I.C., Murphy, T.B., Sweeting, T., Leslie, D.S., Longford, N.T., Kent, J.T., Lawrence, T., Airoldi, E.M., Besag, J., Blei, D., Fienberg, S.E., Breiger, R., Butts, C.T., Doreian, P., Batagelj, V., Ferligoj, A., Draper, D., van Duijn, M.A.J., Faust, K., Petrescu-Prahova, M., Forster, J.J., Gelman, A., Goodreau, S. M., Greenwood, P.E., Gruenberg, K., Francis, B., Hennig, C., Hoff, P.D., Hunter, D.R., Husmeier, D., Glasbey, C., Krackhardt, D., Kuha, J., Skrondal, A., Lawson, A., Liao, T. F., Mendes, B., Reinert, G., Richardson, S., Lewin, A., Titterington, D.M., Wasserman, S., Werhli, A.V. and Ghazal, P.. *Discussion on the paper by Handcock, Raftery and Tantrum.* Journal of the Royal Statistical Society: Series A - Statistics in Society, 170 (2007), pp. 322-354.



p_S -core at level 20 of **Co(SN5)**

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

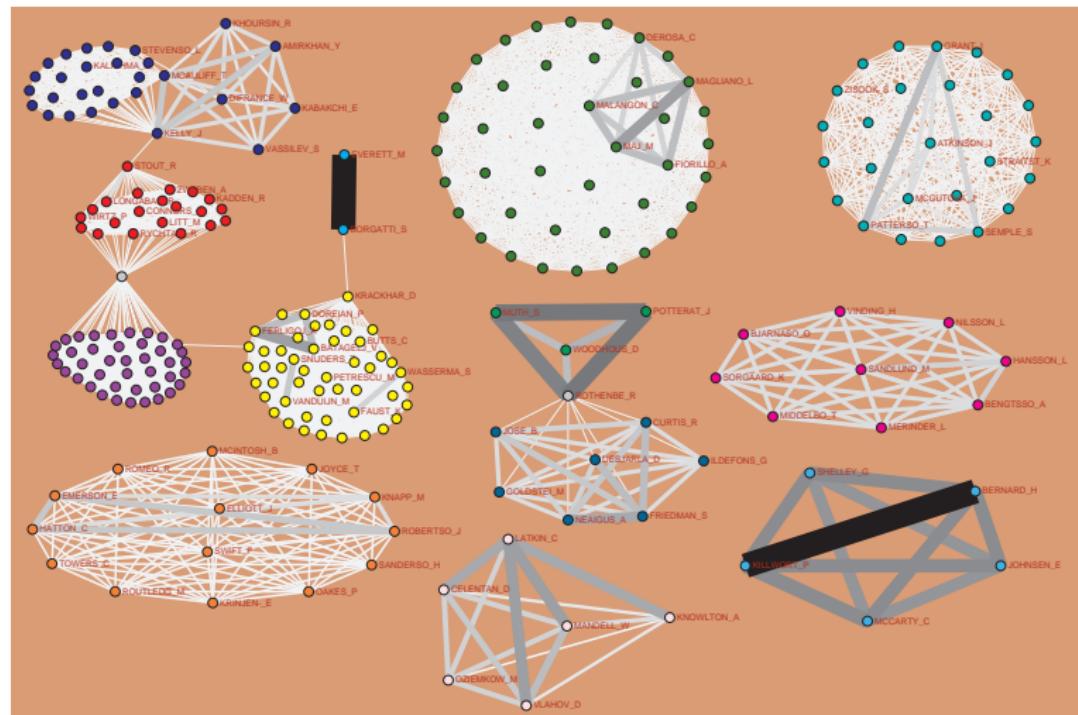
Projections

Collaboration

Other derived networks

EU projects

Temporal Ns





Second co-authorship network

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

$$\mathbf{Cn} = \mathbf{AW} * \mathbf{N}$$

$$cn_{ij} = \sum_{p \in W} wa_{pi} n_{pj} = \sum_{p \in N^-(i) \cap N^-(j)} n_{pj}$$

cn_{ij} = contribution of author j to works, that (s)he wrote together with the author i .

It holds $\sum_{j \in A} \sum_{p \in A} wa_{pi} n_{pj} = \text{outdeg}_{WA}(p)$ and $\sum_{j \in A} cn_{ij} = \text{indeg}_{WA}(i)$

$cn_{ii} = \sum_{p \in N(i)} n_{pi}$ is the contribution of author i to his/her works.

Self-sufficiency: $S_i = \frac{cn_{ii}}{\text{outdeg}_{WA}(i)}$

Collaborativeness: $K_i = 1 - S_i$

$$\sum_{i \in A} \sum_{j \in A} cn_{ij} = \sum_{i \in A} \text{indeg}_{WA}(i) = m_{WA}$$

To compute the table we prepared a macro in **Pajek**.



The "best" authors in Social Networks

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

Collaboration

Other derived networks

EU projects

Temporal Ns

	i	author	cn_{ji}	total	K_j		i	author	cn_{ji}	total	K_j
	1	Burt,R	43.83	53	0.173		26	Latkin,C	10.14	37	0.726
	2	Newman,M	36.77	60	0.387		27	Morris,M	9.98	20	0.501
	3	Doreian,P	34.44	47	0.267		28	Rothenberg,R	9.82	28	0.649
	4	Bonacich,P	30.17	41	0.264		29	Kadushin,C	9.75	11	0.114
	5	Marsden,P	29.42	37	0.205		30	Faust,K	9.72	18	0.460
	6	Wellman,B	26.87	41	0.345		31	Batagelj,V	9.69	20	0.516
	7	Leydesdorf,L	24.37	35	0.304		32	Mizruchi,M	9.67	15	0.356
	8	White,H	23.50	33	0.288		33	[Anon]	9.00	9	0.000
	9	Friedkin,N	20.00	23	0.130		34	Johnson,J	8.89	21	0.577
	10	Borgatti,S	19.20	41	0.532		35	Fararo,T	8.83	16	0.448
	11	Everett,M	16.92	31	0.454		36	Lazega,E	8.50	12	0.292
	12	Litwin,H	16.00	21	0.238		37	Knoke,D	8.33	11	0.242
	13	Freeman,L	15.53	20	0.223		38	Ferligoj,A	8.19	19	0.569
	14	Barabasi,A	14.99	35	0.572		39	Brewer,D	8.03	11	0.270
	15	Snijders,T	14.99	30	0.500		40	Klov Dahl,A	7.96	17	0.532
	16	Valente,T	14.80	34	0.565		41	Hammer,M	7.92	10	0.208
	17	Breiger,R	14.44	20	0.278		42	White,D	7.83	15	0.478
	18	Skvoretz,J	14.43	27	0.466		43	Holme,P	7.42	14	0.470
	19	Krackhardt,D	13.65	25	0.454		44	Boyd,J	7.37	13	0.433
	20	Carley,K	12.93	28	0.538		45	Kilduff,M	7.25	16	0.547
	21	Pattison,P	12.10	27	0.552		46	Small,H	7.00	7	0.000
	22	Wasserman,S	11.72	26	0.549		47	Iacobucci,D	7.00	12	0.417
	23	Berkman,L	11.21	30	0.626		48	Pappi,F	6.83	10	0.317
	24	Moody,J	10.83	15	0.278		49	Chen,C	6.78	12	0.435
	25	Scott,J	10.47	15	0.302		50	Seidman,S	6.75	9	0.250



Third co-authorship network

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

Collaboration

Other derived networks

EU projects

Temporal Ns

$$\mathbf{Ct} = \mathbf{N}^T * \mathbf{N}$$

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

It holds $ct_{ij} = ct_{ji}$ 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.

$\sum_{j \in A} ct_{ij} = \sum_{p \in W} n_{pi}$ = the total contribution of author i to works from W .

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



Components in $\text{Ct}(\text{SN5})$ cut at level 0.5

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

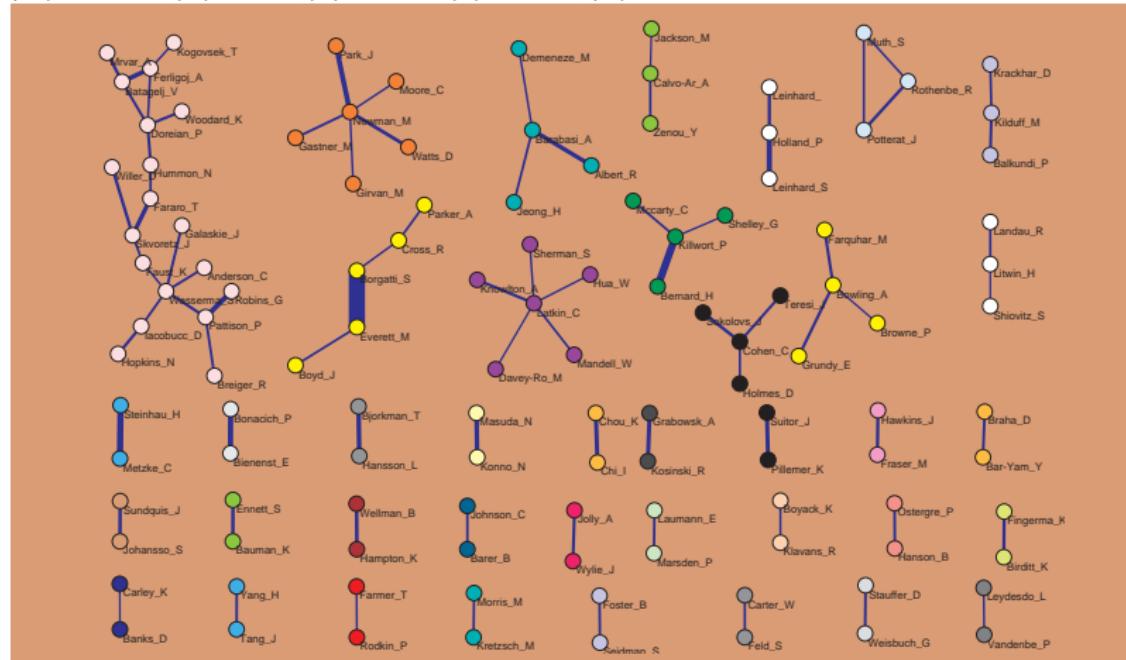
Collaboration

Other derived networks

EU projects

Temporal Ns

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





p_S -core at level 0.75 in $\mathbf{Ct}(\text{SN5})$

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

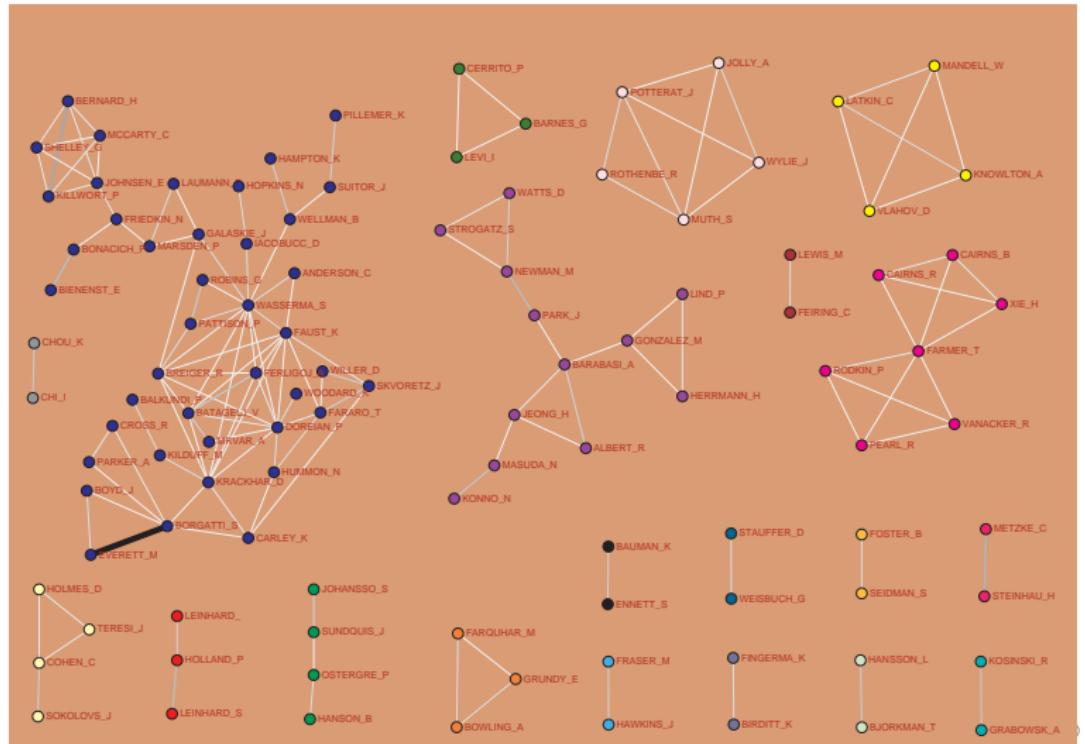
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns



V. Batagelj

Two-mode networks



Some link islands [5,20] in $\text{Ct}(\text{SN5})$

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

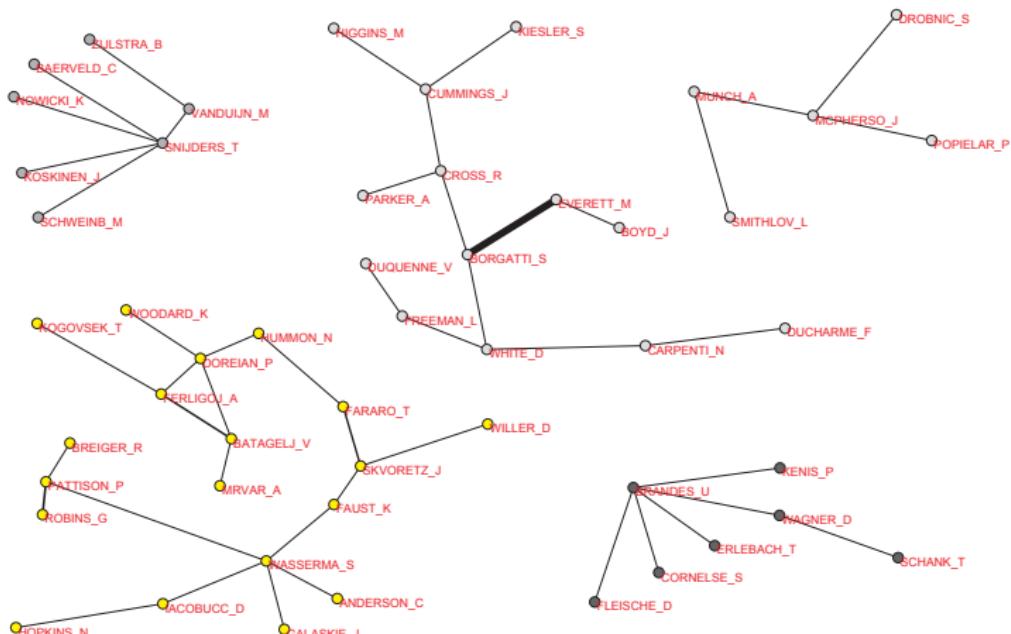
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns





Fourth co-authorship network

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

$$\mathbf{Ct}' = \mathbf{N}^T * \mathbf{N}', \text{ where } n'_{pi} = wa_{pi} / \max(1, \text{outdeg}_{WA}(p) - 1)$$

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

In **Pajek** we can use macros to save sequences of commands to produce different co-authorship networks.

The final result is returned as an undirected simple network with weights (for $i \neq j$)

$$ct'_{ij} = \sum_p \frac{2 \cdot wa_{pi} \cdot wa_{pj}}{\max(1, \text{outdeg}_{WA}(p)) \cdot \max(1, \text{outdeg}_{WA}(p) - 1)}$$



Authors' citations network

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

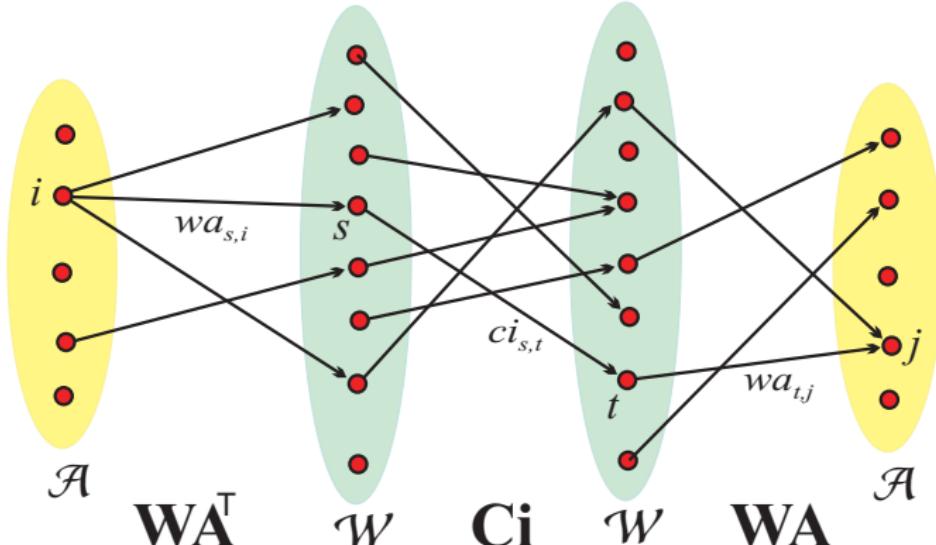
Projections

Collaboration

Other derived networks

EU projects

Temporal Ns



Ca = AW * Ci * WA is a network of citations between authors.
The weight $w(i,j)$ counts the number of times a work authored by i is citing a work authored by j .



Islands in SN5 authors citation network

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

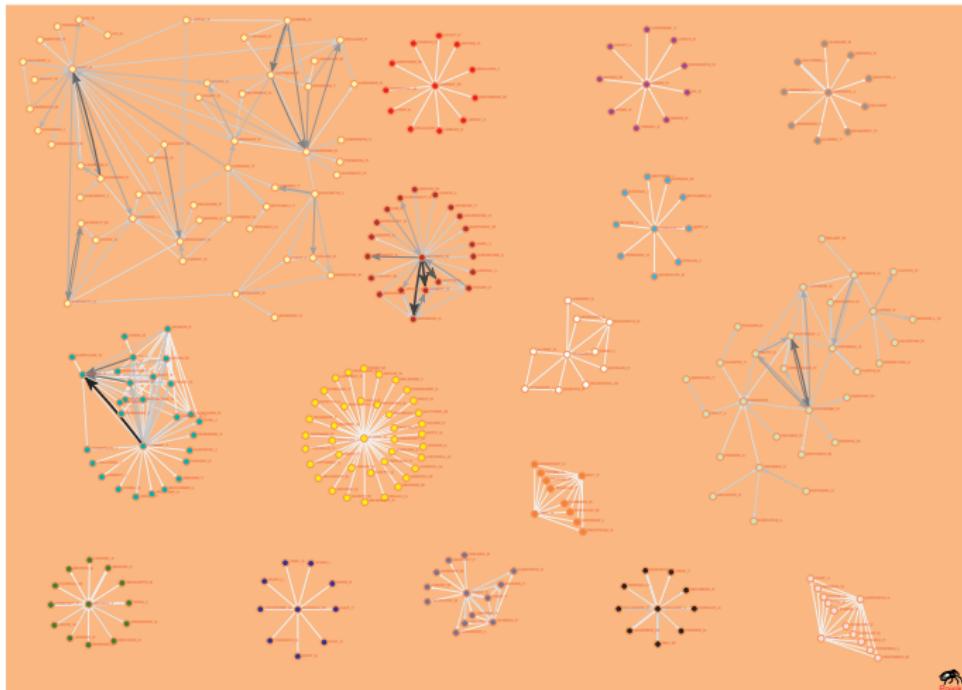
Collaboration

Other derived networks

EU projects

Temporal Ns

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





Bibliographic Coupling

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

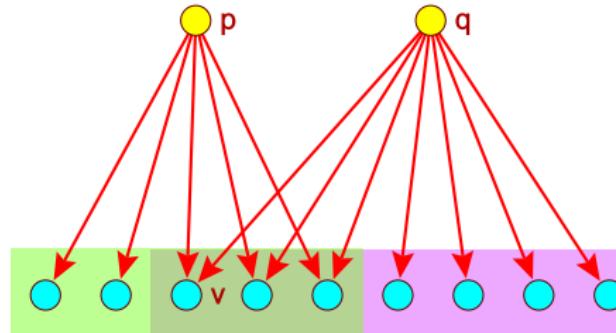
Projections

Collaboration

Other derived networks

EU projects

Temporal Ns



In **WoS2Pajek** the citation relation means $p \text{ Ci } q \equiv$ work p cites work q .

Therefore the *bibliographic coupling* (Kessler, 1963) network **biCo** can be determined as

$$\mathbf{biCo} = \mathbf{Ci} * \mathbf{Ci}^T$$

$bico_{pq} = \# \text{ of works cited by both works } p \text{ and } q = |\mathbf{Ci}(p) \cap \mathbf{Ci}(q)|$.

Bibliographic coupling weights are symmetric: $bico_{pq} = bico_{qp}$:

$$\mathbf{biCo}^T = (\mathbf{Ci} * \mathbf{Ci}^T)^T = \mathbf{Ci} * \mathbf{Ci}^T = \mathbf{biCo}$$



Bibliographic Coupling

fractional approach

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Again we have problems with works with many citations, especially with review papers. To neutralize their impact we can introduce normalized measures. Let's first look at

$$\mathbf{biC} = n(\mathbf{Ci}) * \mathbf{Ci}^T$$

where $n(\mathbf{Ci}) = \mathbf{D} * \mathbf{Ci}$ and $\mathbf{D} = \text{diag}\left(\frac{1}{\max(1, \text{outdeg}(p))}\right)$. $\mathbf{D}^T = \mathbf{D}$.

$$\mathbf{biC} = (\mathbf{D} * \mathbf{Ci}) * \mathbf{Ci}^T = \mathbf{D} * \mathbf{biCo}$$

$$\mathbf{biC}^T = (\mathbf{D} * \mathbf{biCo})^T = \mathbf{biCo}^T * \mathbf{D}^T = \mathbf{biCo} * \mathbf{D}$$

For $\mathbf{Ci}(p) \neq \emptyset$ and $\mathbf{Ci}(q) \neq \emptyset$ it holds (proportions)

$$\mathbf{biC}_{pq} = \frac{|\mathbf{Ci}(p) \cap \mathbf{Ci}(q)|}{|\mathbf{Ci}(p)|} \quad \text{and} \quad \mathbf{biC}_{qp} = \frac{|\mathbf{Ci}(p) \cap \mathbf{Ci}(q)|}{|\mathbf{Ci}(q)|} = \mathbf{biC}_{pq}^T$$

and $\mathbf{biC}_{pq} \in [0, 1]$.



Bibliographic Coupling

fractional approach

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Using **biC** we can construct different normalized measures such as

$$\mathbf{biCoa}_{pq} = \frac{1}{2}(\mathbf{biC}_{pq} + \mathbf{biC}_{qp}) \quad \text{Average}$$

$$\mathbf{biCom}_{pq} = \min(\mathbf{biC}_{pq}, \mathbf{biC}_{qp}) \quad \text{Minimum}$$

or, may be more interesting

$$\mathbf{biCog}_{pq} = \sqrt{\mathbf{biC}_{pq} \cdot \mathbf{biC}_{qp}} = \frac{|\mathbf{Ci}(p) \cap \mathbf{Ci}(q)|}{\sqrt{|\mathbf{Ci}(p)| \cdot |\mathbf{Ci}(q)|}} \quad \begin{matrix} \text{Geometric mean} \\ \text{Salton cosinus} \end{matrix}$$

$$\mathbf{biCoh}_{pq} = 2 \cdot (\mathbf{biC}_{pq}^{-1} + \mathbf{biC}_{qp}^{-1})^{-1} = \frac{2|\mathbf{Ci}(p) \cap \mathbf{Ci}(q)|}{|\mathbf{Ci}(p)| + |\mathbf{Ci}(q)|} \quad \text{Harmonic mean}$$

$$\mathbf{biCoj}_{pq} = (\mathbf{biC}_{pq}^{-1} + \mathbf{biC}_{qp}^{-1} - 1)^{-1} = \frac{|\mathbf{Ci}(p) \cap \mathbf{Ci}(q)|}{|\mathbf{Ci}(p) \cup \mathbf{Ci}(q)|} \quad \text{Jaccard index}$$

All these measures are symmetric.



Bibliographic Coupling

fractional approach

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

biC_{pq} is the proportion of its references the work p shares with the work q .

It is easy to verify that $biCoX_{pq} \in [0, 1]$ and: $biCoX_{pq} = 1$ iff the works p and q are referencing the same works, $\mathbf{Ci}(p) = \mathbf{Ci}(q)$.

From $H \leq G \leq A$ and $J = \frac{H}{2-H}$, $2 - H \geq 1$ we get

$$\mathbf{biCom}_{pq} \leq \mathbf{biCoj}_{pq} \leq \mathbf{biCoh}_{pq} \leq \mathbf{biCog}_{pq} \leq \mathbf{biCoa}_{pq} \leq \mathbf{biCoM}_{pq}$$

The equalities hold iff $\mathbf{Ci}(p) = \mathbf{Ci}(q)$.

To get a dissimilarity use $dis = 1 - sim$ or $dis = \frac{1}{sim} - 1$ or $dis = -\log sim$. For example

$$\mathbf{biCod}_{pq} = 1 - \mathbf{biCoj}_{pq} = \frac{|\mathbf{Ci}(p) \oplus \mathbf{Ci}(q)|}{|\mathbf{Ci}(p) \cup \mathbf{Ci}(q)|} \quad \text{Jaccard distance}$$



Bibliographic Coupling

macro biCon

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

```
select citation network Cite
Network/Create Vector/Centrality/Degree/Output = V1
Vector/Create Constant Vector [n,1] = V2
select V1 as Second vector
Vectors/Max(First,Second)
Vector/Transform/Invert
Network/Create new network/Transform/Transpose 1-mode = CiteT
select network Cite as First
select network CiteT as Second
Networks/Multiply networks = biCo
Operations/Network+Vector/Vector#Network/Output
Network/Create new network/Transform/Remove/Loops = biC
Network/Create new network/Transform/Line values/Power [-1]
Network/Create new network/Transform/Arcs->Edges/Bidirected only/Sum
Network/Create new network/Transform/Line values/Add constant [-1]
Network/Create new network/Transform/Line values/Power [-1] = Jaccard
Network/Create new network/Transform/Line values/Multiply by [-1]
Network/Create new network/Transform/Line values/Add constant [1] = Distance
```



Bibliographic Coupling interpretation

the most cited works from works of a given subnetwork

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

For titles of works from an island see the **CSV files** obtained in R
using the function **description**

```
setwd("C:/Users/batagelj/work/Python/BM/results/jaccard")
source("C:\\\\Users\\\\batagelj\\\\work\\\\Python\\\\WoS\\\\peere1\\\\descript")
T <- read.csv('../titles.csv',sep=";",colClasses="character")
T$code <- 1
dim(T)
d <- description("Jisland4.net","Jisland4.csv",T)
head(d)
d <- description("Jisland7.net","Jisland7.csv",T)
d <- description("Jisland12.net","Jisland12.csv",T)

select Island network as First
select citation network Cite as Second
Networks/Match vertex labels
select partition Positions of Second network in First
Partition/Binarize Partition [1-*]
Partition/Copy to Vector
select transposed network Cite
Operations/Network+Vector/Network*Vector [1]
info Vector [+30]
```



Bibliographic Coupling

the most frequent keywords in works of a given subnetwork

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

```
select Island network as First
select network WK as Second
Networks/Match vertex labels
select partition Positions of Second network in First
Partition/Binarize Partition [1-*]
Partition/Copy to Vector
select WK
Network/Two-mode network/Partition into 2 Modes
Operations/Vector+Partition/Extract Subvector [1]
Network/Two-mode network/Transpose 2-mode
Operations/Network+Vector/Network*Vector [1] = V1
Vector/Constant [n1,0] = V2
select V1 as First
select V2 as Second
Vectors/Fuse vectors
info Vector [+50]
```

The same approach can be applied to WA network.



Co-Citation

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

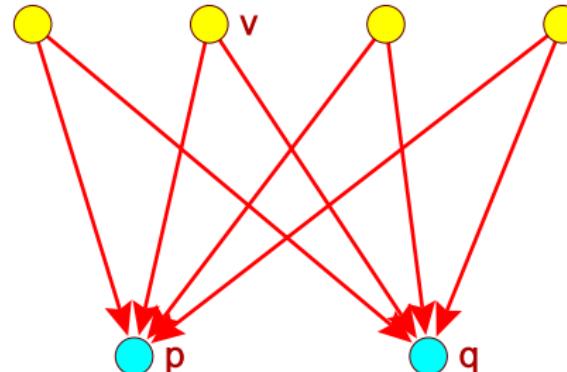
Projections

Collaboration

Other derived networks

EU projects

Temporal Ns



The *co-citation* (Small, Marshakova, 1973) network **coCi** can be determined as

$$\mathbf{coCi} = \mathbf{Ci}^T * \mathbf{Ci}$$

$coci_{pq} = \# \text{ of works citing both works } p \text{ and } q. coci_{pq} = coci_{qp}.$

$$\mathbf{coCi}^T = (\mathbf{Ci}^T * \mathbf{Ci})^T = \mathbf{Ci}^T * \mathbf{Ci} = \mathbf{coCi}$$

$$\begin{aligned} n(\mathbf{Ci})^T * \mathbf{Ci} &= (\mathbf{D} * \mathbf{Ci})^T * \mathbf{Ci} = \mathbf{Ci}^T * (\mathbf{D} * \mathbf{Ci}) \\ &= \mathbf{Ci}^T * n(\mathbf{Ci}) = (n(\mathbf{Ci})^T * \mathbf{Ci})^T \end{aligned}$$

$$\mathbf{CoCin} = n(\mathbf{Ci})^T * \mathbf{Ci}$$



Others

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

The weight $w(a, p)$ in the *author citation* network

$$\mathbf{ACi} = \mathbf{AW} * \mathbf{Ci}$$

counts the number of times author a cited work p .

The *author co-citation* network can be obtained as

$$\mathbf{ACo} = b(\mathbf{ACi}) * t(b(\mathbf{ACi}))$$

Authors using keywords $\mathbf{AK} = \mathbf{AW} * \mathbf{WK}$.



EU projects on simulation

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

For the meeting ***The Age of Simulation*** at Ars Electronica in Linz, January 2006 a dataset of EU projects on simulation was collected by FAS research, Vienna and stored in the form of Excel table (`SimPro.csv`).

The rows are the projects participants (idents) and columns correspond to different their properties. Three two-mode networks were produced from this table using Jürgen Pfeffer's

Text2Pajek program:

- `project.net` – **P** = [idents × projects]
- `country.net` – **C** = [idents × countries]
- `institution.net` – **U** = [idents × institutions]

$|{\text{idents}}| = 8869$, $|{\text{projects}}| = 933$, $|{\text{institutions}}| = 3438$,
 $|{\text{countries}}| = 60$.



EU projects – derived networks

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Since all three networks have the common set (idents) we can derive from them using *network multiplication*

Nets/Multiply First * Second

several interesting networks:

- $\text{ProjInst.net} - \mathbf{W} = [\text{projects} \times \text{institutions}] = \mathbf{P}^T * \mathbf{U}$
- $\text{Countries.net} - \mathbf{S} = [\text{countries} \times \text{countries}] = \mathbf{C}^T * \mathbf{C}$
- $\text{Institutions.net} - \mathbf{Q} = [\text{institutions} \times \text{institutions}]$
 $= \mathbf{W}^T * \mathbf{W}$
- ...

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

Network/2-Mode Network/2-Mode to 1-Mode/Columns



Analysis of ProjInst.net

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

For identifying important parts of ProjInst.net we first computed the 4-rings weights and in the obtained network we determined the line islands

```
Network/Create New Network/With Ring Counts .../4-Rings/Undirect  
Network/Create Partition/Islands/Line Weights[Simple] [2,200]
```

We obtain 101 islands. We extracted 18 islands of the size at least 5. There are two most important islands: aviation companies and car companies.

In labels we used the option \n.



Analysis of ProjInst.net

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

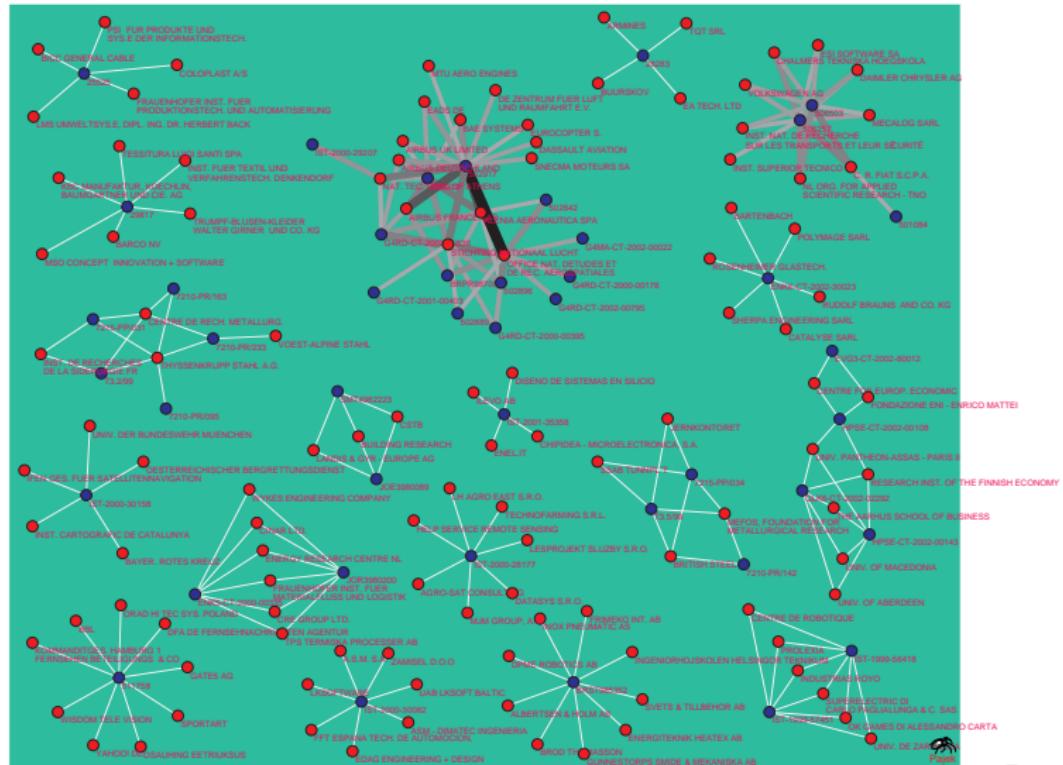
Projections

Collaboration

Other derived networks

EU projects

Temporal Ns





Analysis of Countries.net

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

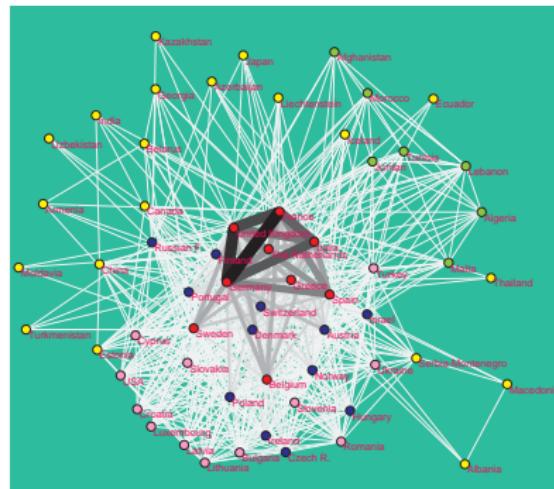
Projections

Collaboration

Other derived networks

EU projects

Temporal Ns



To obtain picture in which the stronger links cover weaker links we have to sort them

Network/Create New

Network/ Transform/Sort
lines/
Line values/Ascending

For dense (sub)networks we get better visualization by using matrix display. In this case we also recoded values (2,10,50).

To determine clusters we used Ward's clustering procedure with dissimilarity measure d_5 (corrected Euclidean distance).

The permutation determined by hierarchy can often be improved by changing the positions of clusters. We get a typical center-periphery structure.



Analysis of Countries.net

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

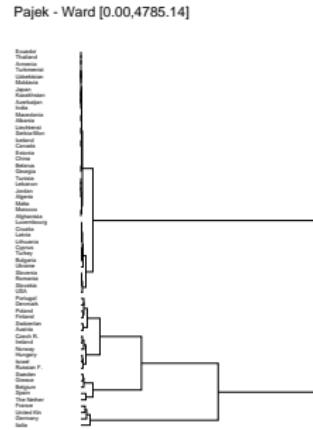
Projections

Collaboration

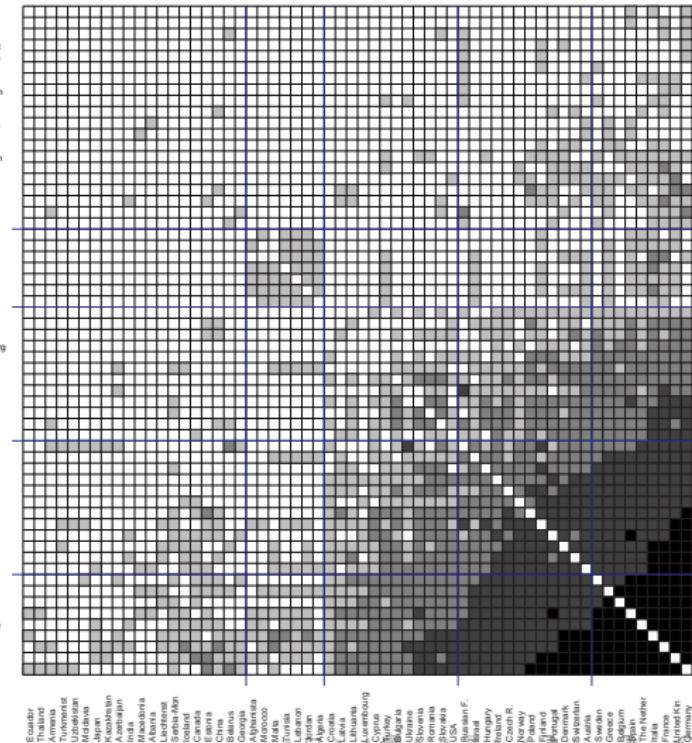
Other derived networks

EU projects

Temporal Ns



Pajek - shadow [0.00,4.00]



V. Batagelj

Two-mode networks





Analysis of Institutions.net

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

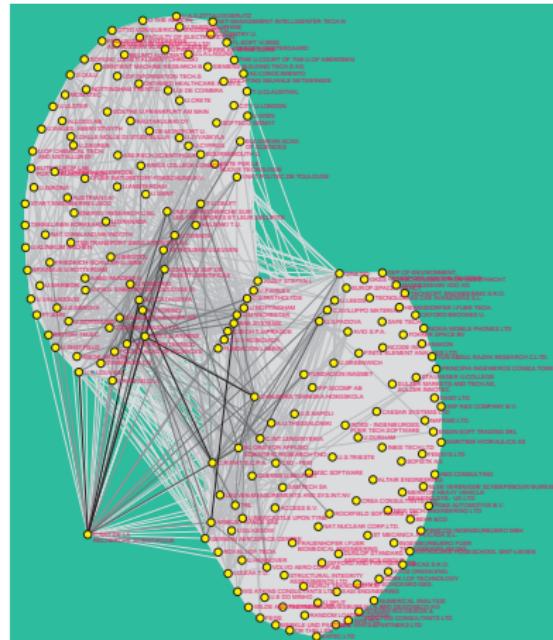
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns



To identify the most important institutions we first computed p_S -cores vector and use it to determine the corresponding node islands. We got essentially one large island. Again the corresponding subnetwork is very dense. We prepared also a matrix display.



Analysis of Institutions.net

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

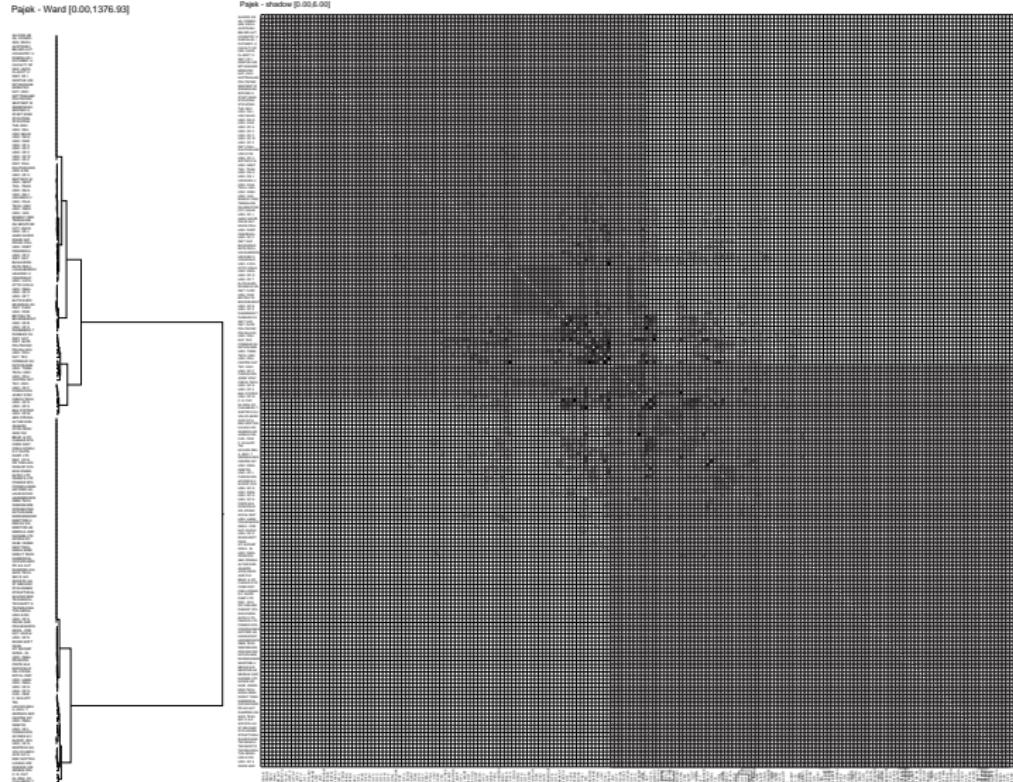
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns



V. Batagelj

Two-mode networks



Temporal network and Levels of analysis

Two-mode networks

V. Batagelj

Two-mode networks

Direct methods

2-mode cores

4-ring weights

Multiplication

Kinship relations

Projections

Collaboration

Other derived networks

EU projects

Temporal Ns

We can also transform the citation network (and other WoS networks) into temporal network using the partition of works by publication year.

Using the time slices also the temporal sequences of corresponding derived networks can be obtained.

Note that most of the obtained derived networks are one-mode networks.

To analyze them standard SNA methods can be used.

In the analysis of the obtained networks the comparability of units could/should be considered.

We are developing a special approach to temporal networks based on temporal quantities. [paper](#)

Pajek allows analyses on different levels specified by a partition of the corresponding set of units and obtained using the *shrinking* of classes.

For example: partition of authors by institutions, or partition of institutions by countries, partitions of authors by discipline/ field/ subfield, etc.

Using the *extraction* of selected classes we can reduce the network to the area of our interest.



Temporal quantities

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

We introduce a notion of a *temporal quantity*

$$a(t) = \begin{cases} a'(t) & t \in T_a \\ \text{\#} & t \in \mathcal{T} \setminus T_a \end{cases}$$

where T_a is the *activity time set* of a and $a'(t)$ is the value of a in an instant $t \in T_a$, and \# denotes the value *undefined*.

We assume that the values of temporal quantities belong to a set A which is a *semiring* $(A, +, \cdot, 0, 1)$ for binary operations $+ : A \times A \rightarrow A$ and $\cdot : A \times A \rightarrow A$.

Let $A_{\text{\#}}(\mathcal{T})$ denote the set of all temporal quantities over $A_{\text{\#}}$ in time \mathcal{T} . To extend the operations to networks and their matrices we first define the *sum* (parallel links) $a + b$ as

$$(a + b)(t) = a(t) + b(t) \quad \text{and} \quad T_{a+b} = T_a \cup T_b.$$

The *product* (sequential links) $a \cdot b$ is defined as

$$(a \cdot b)(t) = a(t) \cdot b(t) \quad \text{and} \quad T_{a \cdot b} = T_a \cap T_b.$$



Sum and product of temporal quantities

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

```
a = [(1, 5, 2), (6, 8, 1), (11, 12, 3), (14, 16, 2),  
      (17, 18, 5), (19, 20, 1)]  
b = [(2, 3, 4), (4, 7, 3), (9, 10, 2), (13, 15, 5),  
      (16, 21, 1)]
```

The following are the sum $s = a + b$ and the product $p = a \cdot b$ of temporal quantities a and b over combinatorial semiring.

```
s = [(1, 2, 2), (2, 3, 6), (3, 4, 2), (4, 5, 5), (5, 6, 3),  
      (6, 7, 4), (7, 8, 1), (9, 10, 2), (11, 12, 3),  
      (13, 14, 5), (14, 15, 7), (15, 16, 2), (16, 17, 1),  
      (17, 18, 6), (18, 19, 1), (19, 20, 2), (20, 21, 1)]  
p = [(2, 3, 8), (4, 5, 6), (6, 7, 3), (14, 15, 10),  
      (17, 18, 5), (19, 20, 1)]
```

They are visually displayed at the bottom half of figures on the following slides.



Addition of temporal quantities.

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

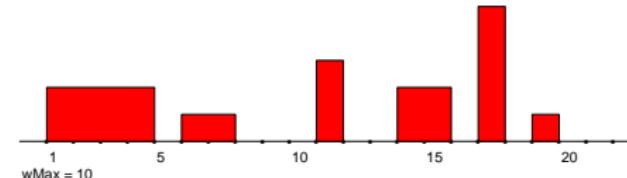
Collaboration

Other derived
networks

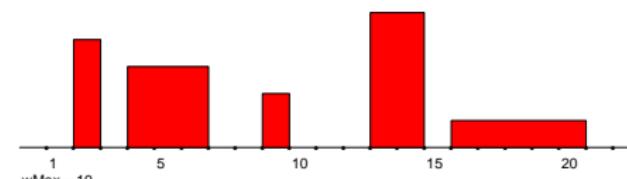
EU projects

Temporal Ns

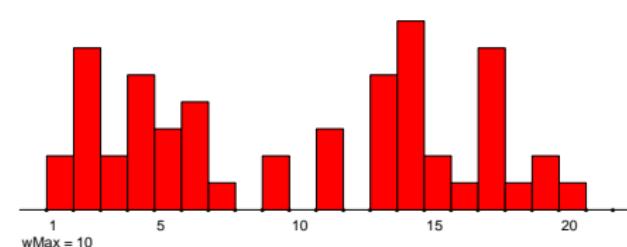
$a :$



$b :$



$a + b :$





Multiplication of temporal quantities.

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

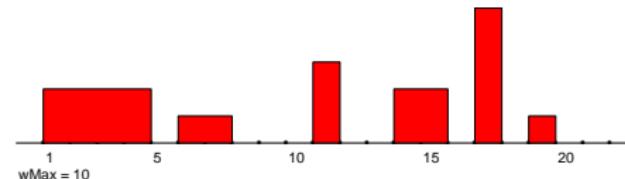
Collaboration

Other derived
networks

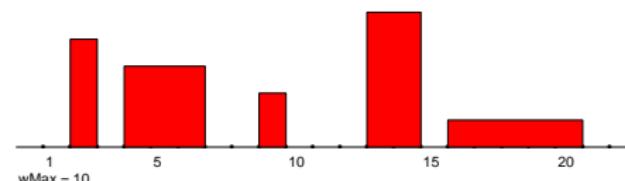
EU projects

Temporal Ns

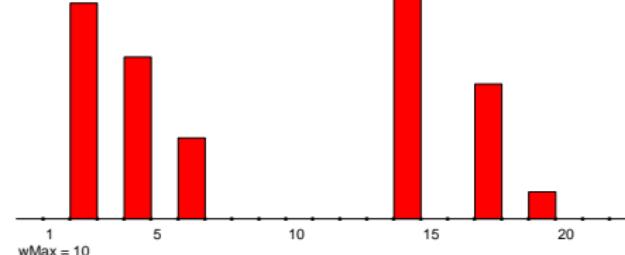
$a :$



$b :$



$a \cdot b :$





Temporal affiliation networks

Let the binary *affiliation* matrix $\mathbf{A} = [a_{ep}]$ describe a two-mode network on the set of events E and the set of participants P :

$$a_{ep} = \begin{cases} 1 & p \text{ participated in the event } e \\ 0 & \text{otherwise} \end{cases}$$

The function $d : E \rightarrow \mathcal{T}$ assigns to each event e the date $d(e)$ when it happened. $\mathcal{T} = [\text{first}, \text{last}] \subset \mathbb{N}$. Using these data we can construct two temporal affiliation matrices:

- **instantaneous $\mathbf{Ai} = [ai_{ep}]$** , where

$$ai_{ep} = \begin{cases} [(d(e), d(e) + 1, 1)] & a_{ep} = 1 \\ [] & \text{otherwise} \end{cases}$$

- **cumulative $\mathbf{Ac} = [ac_{ep}]$** , where

$$ac_{ep} = \begin{cases} [(d(e), last + 1, 1)] & a_{ep} = 1 \\ [] & \text{otherwise} \end{cases}$$



Multiplication of temporal affiliation networks

Instantaneous

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Instantaneous **A** on $P \times A$ and **B** on $P \times B$. **C** = **A**^T.**B** on $A \times B$.

$$c_{ij}(t) = \sum_{p \in P} a_{pi}(t)^T \cdot b_{pj}(t)$$

$a_{pi} = [(d_{pi}, d_{pi} + 1, v_{pi})]$ and $b_{pj} = [(d_{pj}, d_{pj} + 1, v_{pj})]$
for $t = d$ we get

$$c_{ij} = [(d, d + 1, \sum_{p \in P: d_{pi}=d_{pj}=d} v_{pi} \cdot v_{pj})]_{d \in \mathcal{T}}$$

for $v_{pi} = v_{pj} = 1$ we finally get

$$v_{ij}(d) = |\{p \in P : d_{pi} = d_{pj} = d\}|$$

For binary temporal two-mode networks **A** and **B** the value $v_{ij}(d)$ of the product **A**^T.**B** is equal to the number of different members of P with which both i and j have contact in the instant d .



Multiplication of temporal affiliation networks

Cumulative

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Cumulative **A** on $P \times A$ and **B** on $P \times B$. $\mathbf{C} = \mathbf{A}^T \cdot \mathbf{B}$ on $A \times B$.

$$c_{ij}(t) = \sum_{p \in P} a_{pi}(t)^T \cdot b_{pj}(t)$$

$a_{pi} = [(d_{pi}, \text{last} + 1, v_{pi})]$ and $b_{pj} = [(d_{pj}, \text{last} + 1, v_{pj})]$
for $t = d$ we get

$$c_{ij} = [(d, d + 1, \sum_{p \in P : (d_{pi} \leq d) \wedge (d_{pj} \leq d)} v_{pi} \cdot v_{pj})]_{d \in \mathcal{T}}$$

for $v_{pi} = v_{pj} = 1$ we finally get

$$v_{ij}(d) = |\{p \in P : (d_{pi} \leq d) \wedge (d_{pj} \leq d)\}|$$

For binary temporal two-mode networks **A** and **B** the value $v_{ij}(d)$ of the product $\mathbf{A}^T \cdot \mathbf{B}$ is equal to the number of different members of P with which both i and j have contact in all instants up to including the instant d .



Temporal co-authorship networks

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

Using the multiplication of temporal matrices over the combinatorial semiring we get the corresponding instantaneous and cumulative co-occurrence matrices

$$\mathbf{Ci} = \mathbf{Ai}^T \cdot \mathbf{Ai} \quad \text{and} \quad \mathbf{Cc} = \mathbf{Ac}^T \cdot \mathbf{Ac}$$

A typical example of such a matrix is the papers authorship matrix **WA** where E is the set of papers W , P is the set of authors A and d is the publication year.

The triple (s, f, v) in a temporal quantity ci_{pq} tells that in the time interval $[s, f)$ there were v events in which both p and q took part.

The triple (s, f, v) in a temporal quantity cc_{pq} tells that in the time interval $[s, f)$ there were in total v accumulated events in which both p and q took part.

The diagonal matrix entries ci_{pp} and cc_{pp} contain the temporal quantities counting the number of events in the time intervals in which the participant p took part.



Temporal co-authorship network for SN5

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

BibTime

SN5 (2008)

	W	A	K	J
raw	193376	75930	29267	14651
DC=1	7950	12458		

In **Pajek** we extract a subnetwork **WAc** and a corresponding partition **SN5yearC**. Using a program `twoMode2netJSON` we transform them into temporal network in the netJSON format.

Bibliographic networks are usually sparse. The network **WAcInst** has 19488 arcs. The co-authorship network

ColInst = **WAcInst**^T * **WAcInst** has 64980 edges; the corresponding matrix in the package **TQ** has $12458^2 = 155201764$ entries. Using a package **Graph** the co-authorship network is computed in a second and half – a big speed-up.



multiply.py

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

```
gdir = 'c:/users/batagelj/work/python/graph/graph'  
wdir = 'c:/users/batagelj/work/python/graph/JSON/SN5'  
cdir = 'c:/users/batagelj/work/python/graph/chart'  
import sys, os, datetime, json  
sys.path = [gdir]+sys.path; os.chdir(wdir)  
import TQ  
from GraphNew import Graph  
# file = 'C:/Users/batagelj/work/Python/graph/JSON/WAtest.json'  
file = 'C:/Users/batagelj/work/Python/graph/JSON/SN5/WAcInst.json'  
# file = 'C:/Users/batagelj/work/Python/graph/JSON/SN5/WAcCum.json'  
# file = 'C:/Users/batagelj/work/Python/graph/JSON/Gisela/papIns.json'  
t1 = datetime.datetime.now()  
print("started: ",t1.ctime(),"\n")  
G = Graph.loadNetJSON(file)  
t2 = datetime.datetime.now()  
print("\nloaded: ",t2.ctime(),"\ntime used: ", t2-t1)  
# T = G.transpose()  
# Co = Graph.TQmultiply(T,G,True)  
# CR = G.TQtwo2OneRows()  
CC = G.TQtwo2OneCols()  
t3 = datetime.datetime.now()  
print("\ncomputed: ",t3.ctime(),"\ntime used: ", t3-t2)  
ia = { v[3]['lab']: k for k,v in CC._nodes.items() }  
# CC._links[(ia['BORGATTI_S'],ia['EVERETT_M'])][4]['tq']  
# CC._links[(ia['IDI/B'],ia['HCL/B'])][4]['tq']
```



Temporal co-authorship network for SN5

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns

```
===== RESTART: C:\Users\batagelj\work\Python\graph\graph\multiply.py =====
started: Sun Nov 20 00:26:51 2016
loaded: Sun Nov 20 00:26:51 2016
time used: 0:00:00.425024
computed: Sun Nov 20 00:26:52 2016
time used: 0:00:01.165066
>>> BB = CC._links[(ia['BORGATTI_S'],ia['BORGATTI_S'])][4]['tq']
>>> BE = CC._links[(ia['BORGATTI_S'],ia['EVERETT_M'])][4]['tq']
>>> BB
[(1988, 1990, 2), (1990, 1991, 4), (1991, 1992, 2), (1992, 1993, 4),
(1993, 1994, 2), (1994, 1995, 3), (1996, 1997, 1), (1997, 1998, 2),
(1998, 1999, 1), (1999, 2000, 3), (2001, 2002, 2), (2002, 2003, 1),
(2003, 2004, 4), (2005, 2006, 3), (2006, 2007, 2), (2007, 2008, 3)]
>>> BE
[(1988, 1989, 1), (1989, 1990, 2), (1990, 1991, 4), (1991, 1992, 1),
(1992, 1995, 2), (1996, 1998, 1), (1999, 2000, 3), (2003, 2004, 1),
(2005, 2007, 1)]
>>> TQmax = 8; Tmin = 1970; Tmax = 2009; w = 600; h = 120
>>> tit = 'BORGATTI_S'
>>> Graph.TQshow(BB,cdir,TQmax,Tmin,Tmax,w,h,tit,fill='orange')
>>> tit = 'BORGATTI_S - EVERETT_M'
>>> Graph.TQshow(BE,cdir,TQmax,Tmin,Tmax,w,h,tit,fill='orange')
>>> NN = CC._links[(ia['NEWMAN_M'],ia['NEWMAN_M'])][4]['tq']
>>> NN
[(1999, 2000, 2), (2000, 2001, 4), (2001, 2002, 7), (2002, 2003, 8),
(2003, 2004, 7), (2004, 2005, 11), (2005, 2006, 7), (2006, 2007, 11),
(2007, 2008, 3)]
>>> tit = 'NEWMAN_M'; TQmax = 12; h = 150
>>> Graph.TQshow(NN,cdir,TQmax,Tmin,Tmax,w,h,tit,fill='orange')
```



Visualization

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

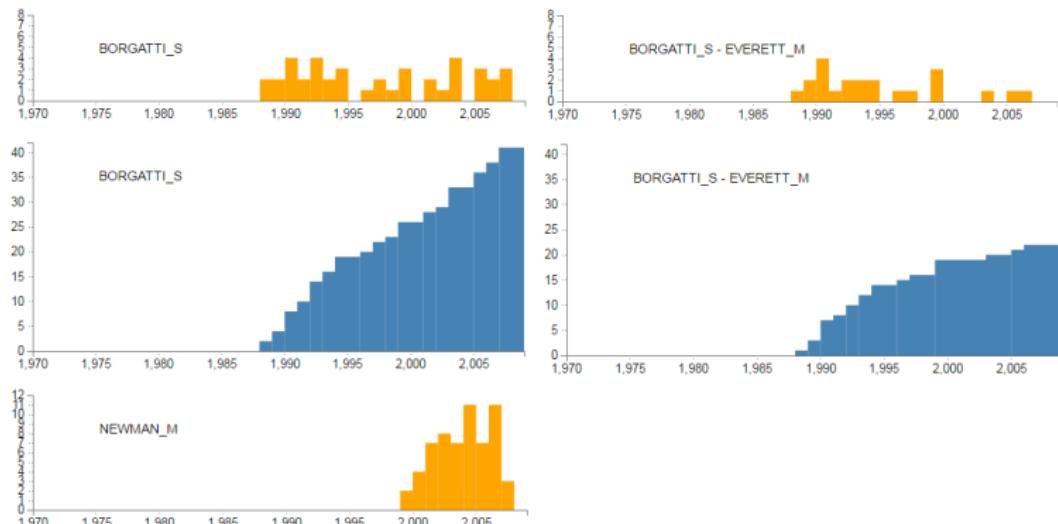
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns





Understanding large networks

Two-mode
networks

V. Batagelj

Two-mode
networks

Direct
methods

2-mode cores

4-ring weights

Multiplication

Kinship
relations

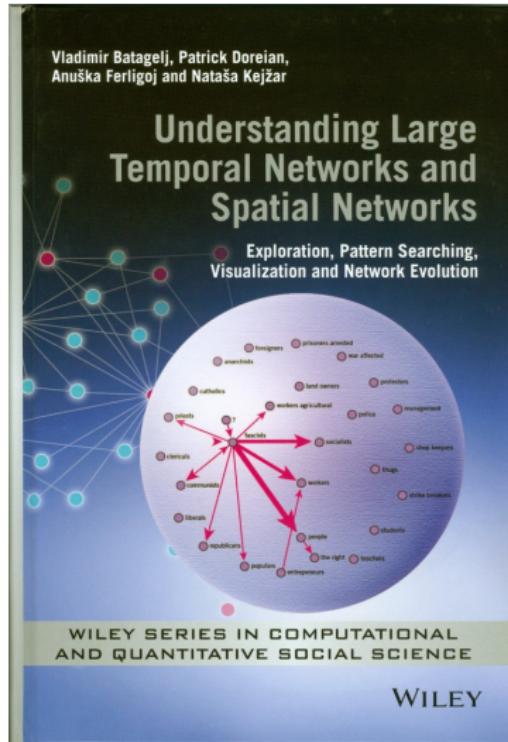
Projections

Collaboration

Other derived
networks

EU projects

Temporal Ns



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.



Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Introduction to Network Analysis using **Pajek**

8. Clustering and Blockmodeling

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

PhD and MS program in Statistics
University of Ljubljana, 2022



Outline

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

- 1 Clustering**
- 2 Block modeling**
- 3 Generalized Blockmodeling**
- 4 Pre-specified blockmodeling**
- 5 Two-mode blockmodeling**
- 6 Signed graphs**
- 7 Generalizations**
- 8 Clustering with constraints**

TYPE OF MEMBERSHIP	Max. size	EVENTS AND PARTICIPATIONS													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
<i>Clique I:</i>	1	C	C	C	C	C	C	-	C	C					
Core.....	2	C	C	C	-	C	C	C	C	C	-				
	3	-	C	C	C	C	C	C	C	C	C	-			
	4	C	-	C	C	C	C	C	C	C	C	-			
	5			P	P	P	-	P	-	-	-				
Primary...	6			P	-	P	P	-	P	-					
	7				P	P	P	P	P	P	-				
Secondary.	8				-	S	-	S	S	S					
<i>Clique II:</i>	9				S	-	S	S	S	S					
Secondary.	10					S	S	S	S	-	-	S			
	11					-	P	P	P	-	P	P			
	12					-	P	P	P	-	P	P	P		
	13					C	C	C	C	-	C	C	C		
Core.....	14					C	-	C	C	C	C	C	C		
	15					C	C	-	C	C	C	C	C	C	
Secondary.	16					S	S	[S - S]	S	S					
	17					S	-	S	S	S					
	18					S	-	S	S	S					

Figure 2. Participation of the Southern Women in Events

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

Current version of slides (March 23, 2022 at 00:41): [slides PDF](#)



Clustering problem

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Let us start with the formal setting of the clustering problem. We shall use the following notation:

X – *unit*

X – *description* of unit X

\mathcal{U} – *space* of units

\mathcal{U} – finite *set of units*, $\mathcal{U} \subset \mathcal{U}$

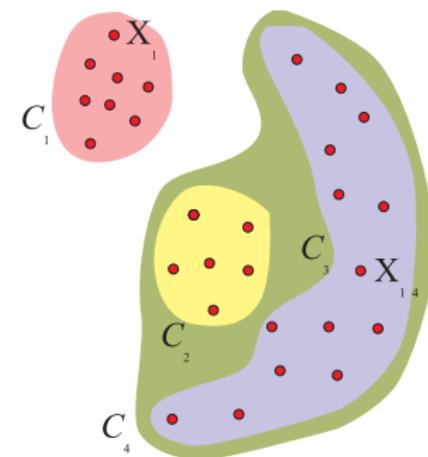
C – *cluster*, $\emptyset \subset C \subseteq \mathcal{U}$

\mathbf{C} – *clustering*, $\mathbf{C} = \{C_i\}$

Φ – set of *feasible clusterings*

P – *criterion function*,

$$P : \Phi \rightarrow \mathbb{R}_0^+$$





... Clustering problem

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

With these notions we can express the *clustering problem* (Φ, P) as follows:

Determine the clustering $\mathbf{C}^ \in \Phi$ for which*

$$P(\mathbf{C}^*) = \min_{\mathbf{C} \in \Phi} P(\mathbf{C})$$

Since the set of units \mathcal{U} is finite, the set of feasible clusterings is also finite. Therefore the set $\text{Min}(\Phi, P)$ of all solutions of the problem (optimal clusterings) is not empty. (In theory) the set $\text{Min}(\Phi, P)$ can be determined by the complete search.

We shall denote the value of criterion function for an optimal clustering by $\min(\Phi, P)$.

Generally the clusters of clustering $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$ need not to be pairwise disjoint; yet, the clustering theory and practice mainly deal with clusterings which are the *partitions* of \mathcal{U} . We shall denote the set of all partitions of \mathcal{U} into k classes (clusters) by $\Pi_k(\mathcal{U})$.



Simple criterion functions

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Joining the individual units into a cluster C we make a certain "error", we create certain "tension" among them – we denote this quantity by $p(C)$. The *criterion function* $P(\mathbf{C})$ combines these "partial/local errors" into a "global error".

Usually it takes the form:

$$\text{S.} \quad P(\mathbf{C}) = \sum_{C \in \mathbf{C}} p(C) \quad \text{or} \quad \text{M.} \quad P(\mathbf{C}) = \max_{C \in \mathbf{C}} p(C)$$

For simple criterion functions usually
 $\min(\Pi_{k+1}\mathcal{U}, P) \leq \min(\Pi_k(\mathcal{U}), P)$
we fix the value of k and set $\Phi \subseteq \Pi_k(\mathcal{U})$.



Cluster-error function / dissimilarities

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The *cluster-error* $p(C)$ has usually the properties:

$$p(C) \geq 0 \quad \text{and} \quad \forall X \in \mathcal{U} : p(\{X\}) = 0$$

In the following we shall assume that these properties of $p(C)$ hold.

To express the cluster-error $p(C)$ we define on the space of units a *dissimilarity* $d : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}_0^+$ for which we require:

D1. $\forall X \in \mathcal{U} : d(X, X) = 0$

D2. *symmetric*: $\forall X, Y \in \mathcal{U} : d(X, Y) = d(Y, X)$

Usually the dissimilarity d is defined using another dissimilarity $\delta : [\mathcal{U}] \times [\mathcal{U}] \rightarrow \mathbb{R}_0^+$ as

$$d(X, Y) = \delta([X], [Y])$$



Cluster-error function / examples

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Now we can define several cluster-error functions:

$$S. \quad p(C) = \sum_{X, Y \in C, X < Y} w(X) \cdot w(Y) \cdot d(X, Y)$$

$$\bar{S}. \quad p(C) = \frac{1}{w(C)} \sum_{X, Y \in C, X < Y} w(X) \cdot w(Y) \cdot d(X, Y)$$

where $w : \mathcal{U} \rightarrow \mathbb{R}^+$ is a *weight* of units, which is extended to clusters by:

$$w(\{X\}) = w(X), \quad X \in \mathcal{U}$$

$$w(C_1 \cup C_2) = w(C_1) + w(C_2), \quad C_1 \cap C_2 = \emptyset$$

Often $w(X) = 1$ holds for each $X \in \mathcal{U}$. Then $w(C) = \text{card}(C)$.

$$M. \quad p(C) = \max_{X, Y \in C} d(X, Y) = \text{diam}(C) \quad - \text{diameter}$$

$$T. \quad p(C) = \min_{T \text{ is a spanning tree over } C} \sum_{(X:Y) \in T} d(X, Y)$$



Matrix rearrangement view on blockmodeling

Snyder & Kick's World trade network / $n = 118$, $m = 514$

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

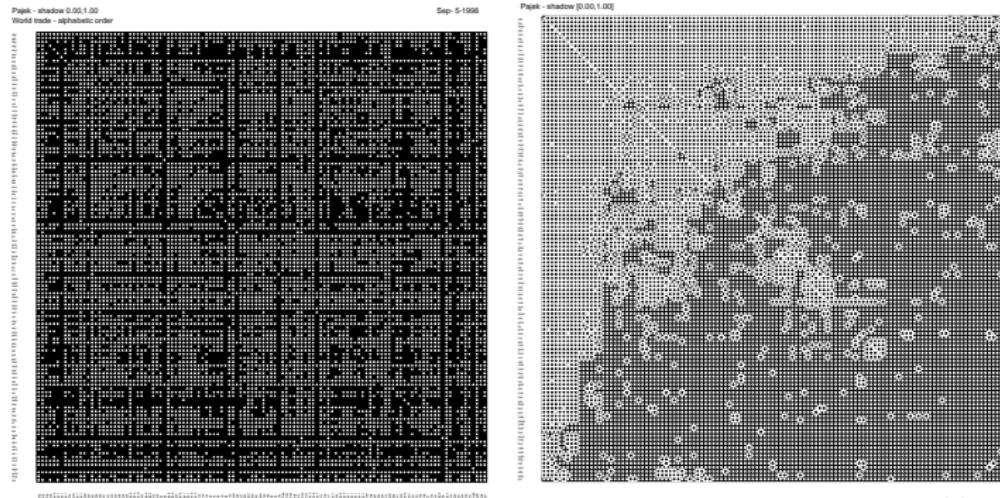
Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints



Alphabetic order of countries (left) and rearrangement (right)



Ordering the matrix

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

There are several ways how to rearrange a given matrix – determine an *ordering* or *permutation* of its rows and columns – to get some insight into its structure. For example:

- ordering by degree; by connected components; or by core number, connected components inside core levels, and degree;
- ordering according to a hierarchical clustering and some other property.

There exists also some special procedures to determine the ordering such as seriation and clumping (Murtagh) or **RCM** – Reverse Cuthill-McKee algorithm.

Partition/Make Permutation

Vector/Make Permutation

Network/Create Permutation/Reverse Cuthill-McKee

Network/Create Permutation/Core+Degree



Blockmodeling as a clustering problem

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

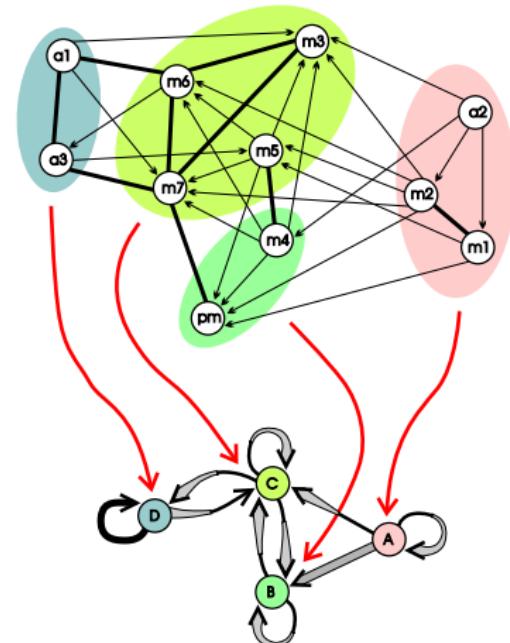
Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The goal of *blockmodeling* is to reduce a large, potentially incoherent network to a smaller comprehensible structure that can be interpreted more readily. Blockmodeling, as an empirical procedure, is based on the idea that units in a network can be grouped according to the extent to which they are equivalent, according to some *meaningful* definition of equivalence.





Cluster, clustering, blocks

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

One of the main procedural goals of blockmodeling is to identify, in a given network $\mathcal{N} = (\mathcal{U}, R)$, $R \subseteq \mathcal{U} \times \mathcal{U}$, **clusters** (classes) of units that share structural characteristics defined in terms of R .

The units within a cluster have the same or similar connection patterns to other units. They form a **clustering**

$\mathbf{C} = \{C_1, C_2, \dots, C_k\}$ which is a **partition** of the set \mathcal{U} . Each partition determines an equivalence relation (and vice versa). Let us denote by \sim the relation determined by partition **C**.

A clustering **C** partitions also the relation R into **blocks**

$$R(C_i, C_j) = R \cap C_i \times C_j$$

Each such block consists of units belonging to clusters C_i and C_j and all arcs leading from cluster C_i to cluster C_j . If $i = j$, a block $R(C_i, C_i)$ is called a **diagonal** block.



Structural and regular equivalence

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Regardless of the definition of equivalence used, there are two basic approaches to the equivalence of units in a given network (compare Faust, 1988):

- the equivalent units have the same connection pattern to the **same** neighbors;
- the equivalent units have the same or similar connection pattern to (possibly) **different** neighbors.

The first type of equivalence is formalized by the notion of structural equivalence and the second by the notion of regular equivalence with the latter a generalization of the former.



Structural equivalence

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Units are equivalent if they are linked to the rest of the network in *identical* ways (Lorrain and White, 1971). Such units are said to be *structurally equivalent*.

The units X and Y are *structurally equivalent*, we write $X \equiv Y$, iff the permutation (transposition) $\pi = (XY)$ is an automorphism of the relation R (Borgatti and Everett, 1992).

In other words, X and Y are structurally equivalent iff:

- | | |
|-------------------------------|--|
| s1. $XRY \Leftrightarrow YRX$ | s3. $\forall Z \in \mathcal{U} \setminus \{X, Y\} : (XRZ \Leftrightarrow YRZ)$ |
| s2. $XRX \Leftrightarrow YRY$ | s4. $\forall Z \in \mathcal{U} \setminus \{X, Y\} : (ZRX \Leftrightarrow ZRY)$ |



... structural equivalence

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0



Regular equivalence

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Integral to all attempts to generalize structural equivalence is the idea that units are equivalent if they link in equivalent ways to other units that are also equivalent.

White and Reitz (1983): The equivalence relation \approx on \mathcal{U} is a *regular equivalence* on network $\mathcal{N} = (\mathcal{U}, R)$ if and only if for all $X, Y, Z \in \mathcal{U}$, $X \approx Y$ implies both

$$R1. \quad XRZ \Rightarrow \exists W \in \mathcal{U} : (YRW \wedge W \approx Z)$$

$$R2. \quad ZRX \Rightarrow \exists W \in \mathcal{U} : (WRY \wedge W \approx Z)$$

Another view of regular equivalence is based on colorings (Everett, Borgatti 1996): regular equivalent nodes are of the same color and have the same set of colors in their neighborhoods.

$$X \approx Y \Leftrightarrow (b(X) = b(Y)) \wedge (b(N(X)) = b(N(Y)))$$



... regular equivalence

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Theorem (Batagelj, Doreian, Ferligoj, 1992)

Let $\mathbf{C} = \{C_i\}$ be a partition corresponding to a regular equivalence \approx on the network $\mathcal{N} = (\mathcal{U}, R)$. Then each block $R(C_u, C_v)$ is either null or it has the property that there is at least one 1 in each of its rows and in each of its columns. Conversely, if for a given clustering \mathbf{C} , each block has this property then the corresponding equivalence relation is a regular equivalence.

The blocks for regular equivalence are null or 1-covered blocks.

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

1	0	1	0	0
0	0	1	0	1
0	1	0	0	0
1	0	1	1	0



Establishing Blockmodels

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The problem of establishing a partition of units in a network in terms of a selected type of equivalence is a special case of *clustering problem* (Φ, P):

Determine the clustering $\mathbf{C}^ \in \Phi$ for which*

$$P(\mathbf{C}^*) = \min_{\mathbf{C} \in \Phi} P(\mathbf{C})$$

where Φ is the set of *feasible clusterings* and P is a *criterion function*.

Criterion functions can be constructed

- *indirectly* as a function of a compatible (dis)similarity measure between pairs of units, or
- *directly* as a function measuring the fit of a clustering to an ideal one with perfect relations within each cluster and between clusters according to the considered types of connections (equivalence).



Indirect approach

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

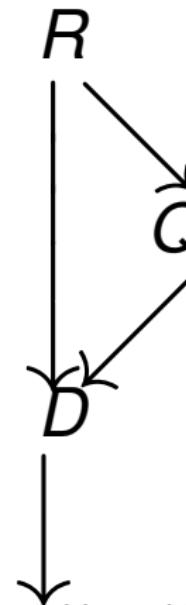
Clustering
with
constraints

RELATION

DESCRIPTIONS
OF UNITS

DISSIMILARITY
MATRIX

STANDARD
CLUSTERING
ALGORITHMS



original relation

path matrix
triads
orbits

hierarchical algorithms,
relocation algorithm, leader algorithm, etc.



Dissimilarities

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The property $t : \mathcal{U} \rightarrow \mathbb{R}$ is *structural property* if, for every automorphism φ , of the relation R , and every unit $x \in \mathcal{U}$, it holds that $t(x) = t(\varphi(x))$. Some examples of a structural property include

$t(u)$ = the *degree* of unit u ;

$t(u)$ = number of units at *distance* d from the unit u ;

$t(u)$ = number of *triads* of type x at the unit u .

Centrality measures are further examples of structural properties.

We can define the description of the unit u as

$[u] = [t_1(u), t_2(u), \dots, t_m(u)]$. As a simple example, t_1 could be *degree* centrality, t_2 could be *closeness* centrality and t_3 could be *betweenness* centrality. The dissimilarity between units u and v could be defined as $d(u, v) = D([u], [v])$ where D is some (standard) dissimilarity between real vectors. In the simple example, D could be the *Euclidean* distance between the centrality profiles.



Dissimilarities based on matrices

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

We consider the following list of dissimilarities between units x_i and x_j where the description of the unit consists of the row and column of the property matrix $\mathbf{Q} = [q_{ij}]$. We take as units the rows of the matrix

$$\mathbf{X} = [\mathbf{Q}\mathbf{Q}^T]$$



... Dissimilarities

Manhattan distance: $d_m(x_i, x_j) = \sum_{s=1}^n (|q_{is} - q_{js}| + |q_{si} - q_{sj}|)$

Euclidean distance:

$$d_E(x_i, x_j) = \sqrt{\sum_{s=1}^n ((q_{is} - q_{js})^2 + (q_{si} - q_{sj})^2)}$$

Truncated Manhattan distance:

$$d_s(x_i, x_j) = \sum_{\substack{s=1 \\ s \neq i,j}}^n (|q_{is} - q_{js}| + |q_{si} - q_{sj}|)$$

Truncated Euclidean distance (Faust, 1988):

$$d_S(x_i, x_j) = \sqrt{\sum_{\substack{s=1 \\ s \neq i,j}}^n ((q_{is} - q_{js})^2 + (q_{si} - q_{sj})^2)}$$



... Dissimilarities

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Corrected Manhattan-like dissimilarity ($p \geq 0$):

$$d_c(p)(x_i, x_j) = d_s(x_i, x_j) + p \cdot (|q_{ii} - q_{jj}| + |q_{ij} - q_{ji}|)$$

Corrected Euclidean-like dissimilarity (Burt and Minor, 1983):

$$d_e(p)(x_i, x_j) = \sqrt{d_s(x_i, x_j)^2 + p \cdot ((q_{ii} - q_{jj})^2 + (q_{ij} - q_{ji})^2)}$$

Corrected dissimilarity:

$$d_c(p)(x_i, x_j) = \sqrt{d_c(p)(x_i, x_j)}$$

The parameter, p , can take any positive value. Typically, $p = 1$ or $p = 2$, where these values count the number of times the corresponding diagonal pairs are counted.

Operations/Network+Cluster/Dissimilarity*/Network based
Network/Create Hierarchy/Clustering*/



... Dissimilarities

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

It is easy to verify that all expressions from the list define a dissimilarity (i.e. that $d(x, y) \geq 0$; $d(x, x) = 0$; and $d(x, y) = d(y, x)$). Each of the dissimilarities from the list can be assessed to see whether or not it is also a distance:
 $d(x, y) = 0 \Rightarrow x = y$ and $d(x, y) + d(y, z) \geq d(x, z)$.

The dissimilarity measure d is *compatible* with a considered equivalence \sim if for each pair of units holds

$$X_i \sim X_j \Leftrightarrow d(X_i, X_j) = 0$$

Not all dissimilarity measures typically used are compatible with structural equivalence. For example, the *corrected Euclidean-like dissimilarity* is compatible with structural equivalence.

The indirect clustering approach does not seem suitable for establishing clusterings in terms of regular equivalence since there is no evident way how to construct a compatible (dis)similarity measure.



Example: Support network among informatics students

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The analyzed network consists of social support exchange relation among fifteen students of the Social Science Informatics fourth year class (2002/2003) at the Faculty of Social Sciences, University of Ljubljana. Interviews were conducted in October 2002.

Support relation among students was identified by the following question:

Introduction: You have done several exams since you are in the second class now. Students usually borrow studying material from their colleagues.

Enumerate (list) the names of your colleagues that you have most often borrowed studying material from. (The number of listed persons is not limited.)



Class network

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

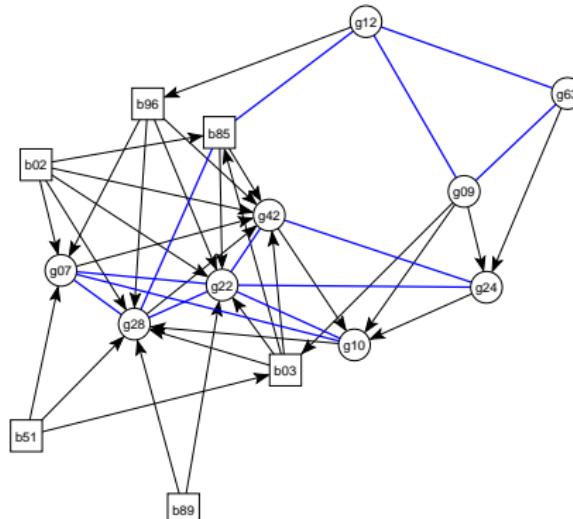
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



class.net

Nodes represent students in the class; circles – girls, squares – boys. Opposite pairs of arcs are replaced by edges.

Cluster/Create Complete Cluster [n]

Operations/Network+Cluster/Dissimilarity*/Network Based/



Indirect approach

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

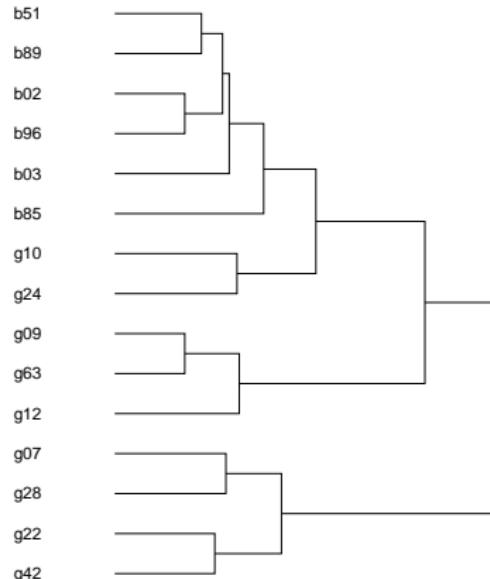
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



Using *Corrected Euclidean-like dissimilarity* and *Ward clustering method* we obtain the following dendrogram.

From it we can determine the number of clusters: 'Natural' clusterings correspond to clear 'jumps' in the dendrogram.
If we select 3 clusters we get the partition **C**.

$$\mathbf{C} = \{\{b51, b89, b02, b96, b03, b85, g10, g24\}, \\ \{g09, g63, g12\}, \{g07, g28, g22, g42\}\}$$



Partition in 3 clusters

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

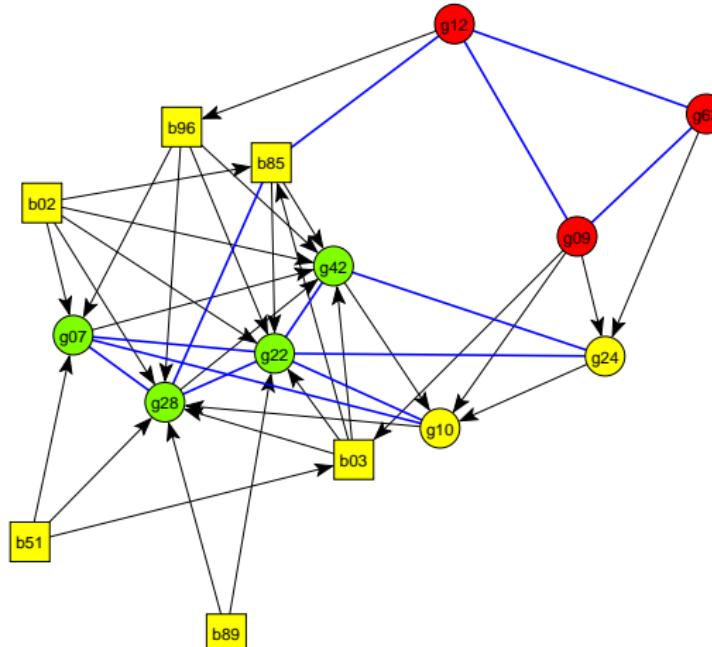
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



On the picture, nodes in the same cluster are of the same color.



Matrix

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

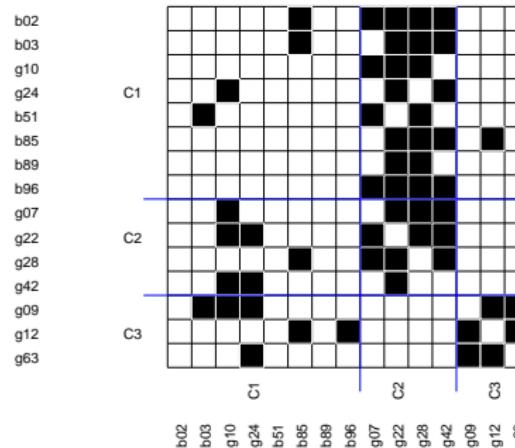
Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

Pajek - shadow [0.00,1.00]



The partition can be used also to reorder rows and columns of the matrix representing the network. Clusters are divided using blue vertical and horizontal lines.

select cluster in the hierarchy

Hierarchy/Make permutation

Hierarchy/Make partition

File/Network/Export as Matrix to EPS/Using Permutation



Direct Approach and Generalized Blockmodeling

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The second possibility for solving the blockmodeling problem is to construct an appropriate criterion function directly and then use a local optimization algorithm to obtain a ‘good’ clustering solution. Criterion function $P(\mathbf{C})$ has to be *sensitive* to considered equivalence:

$$P(\mathbf{C}) = 0 \Leftrightarrow \mathbf{C} \text{ defines considered equivalence.}$$



Generalized Blockmodeling

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

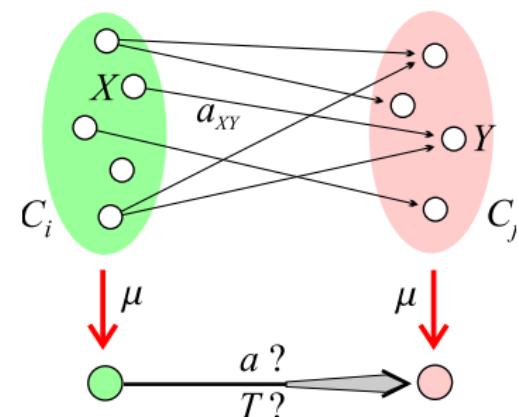
Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

A *blockmodel* consists of structures obtained by identifying all units from the same cluster of the clustering \mathbf{C} . For an exact definition of a blockmodel we have to be precise also about which blocks produce an arc in the *reduced graph* and which do not, and of what *type*. Some types of connections are presented in the figure on the next slide. The reduced graph can be represented by relational matrix, called also *image matrix*.





Block Types

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

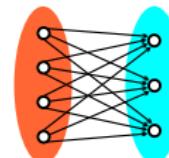
Two-mode
blockmodeling

Signed graphs

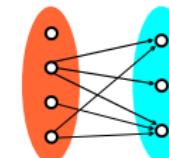
Generalizations

Clustering
with
constraints

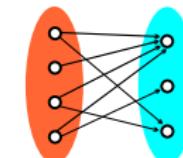
complete



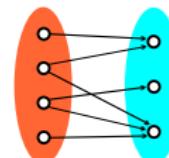
row-dominant



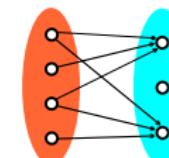
col-dominant



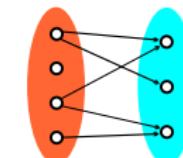
regular



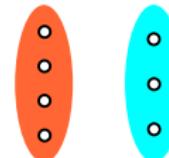
row-regular



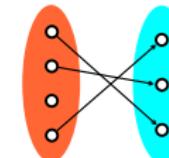
col-regular



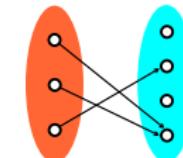
null



row-functional



col-functional





Generalized equivalence / Block Types

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

	Y		Y		Y	
X	1 1 1 1 1		0 1 0 0 0		0 0 1 0 0	
	1 1 1 1 1		1 1 1 1 1		0 0 1 1 0	
	1 1 1 1 1		0 0 0 0 0		1 1 1 0 0	
	1 1 1 1 1		0 0 0 1 0		0 0 1 0 1	
	complete			row-dominant		
X	0 1 0 0 0		0 1 0 0 0		0 1 0 1 0	
	1 0 1 1 0		0 1 1 0 0		1 0 1 0 0	
	0 0 1 0 1		1 0 1 0 0		1 1 0 1 1	
	1 1 0 0 0		0 1 0 0 1		0 0 0 0 0	
	regular			row-regular		
X	0 0 0 0 0		0 0 0 1 0		1 0 0 0 0	
	0 0 0 0 0		0 0 1 0 0		0 1 0 0 0	
	0 0 0 0 0		1 0 0 0 0		0 0 1 0 0	
	0 0 0 0 0		0 0 0 1 0		0 0 0 0 1	
	null			row-functional		
X	1 0 0 0 0		0 1 0 0 0		0 0 1 0 0	
	0 1 0 0 0		0 0 1 0 0		0 0 0 1 0	
	0 0 1 0 0		1 0 0 0 0		0 0 0 0 1	
	0 0 0 1 0		0 0 0 1 0		col-functional	



Characterizations of Types of Blocks

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

null	nul	all 0 *	
complete	com	all 1 *	
regular	reg	1-covered rows and columns	
row-regular	rre	each row is 1-covered	
col-regular	cre	each column is 1-covered	
row-dominant	rdo	\exists all 1 row *	
col-dominant	cdo	\exists all 1 column *	
row-functional	rfn	$\exists!$ one 1 in each row	
col-functional	cfn	$\exists!$ one 1 in each column	
non-null	one	\exists at least one 1	

* except this may be diagonal



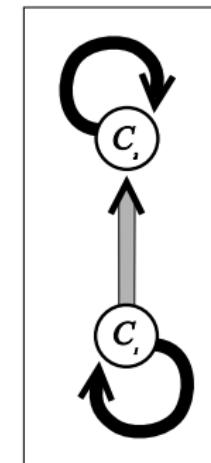
Block Types and Matrices

Clustering

V. Batagelj

1	1	1	1	1	1	0	0
1	1	1	1	0	1	0	1
1	1	1	1	0	0	1	0
1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1
0	0	0	0	1	0	1	1
0	0	0	0	1	1	0	1
0	0	0	0	1	1	1	0

	C_1	C_2
C_1	complete	regular
C_2	null	complete





Formalization of blockmodeling

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Let V be a set of positions or images of clusters of units. Let $\mu : \mathcal{U} \rightarrow V$ denote a mapping which maps each unit to its position. The cluster of units $C(t)$ with the same position $t \in V$ is

$$C(t) = \mu^{-1}(t) = \{X \in \mathcal{U} : \mu(X) = t\}$$

Therefore

$$\mathbf{C}(\mu) = \{C(t) : t \in V\}$$

is a partition (clustering) of the set of units \mathcal{U} .



Blockmodel

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

A *blockmodel* is an ordered sextuple $\mathcal{M} = (V, K, \mathcal{T}, Q, \pi, \alpha)$ where:

- V is a set of *types of units* (images or representatives of classes);
- $K \subseteq V \times V$ is a set of *connections*;
- \mathcal{T} is a set of predicates used to describe the *types of connections* between different classes (clusters, groups, types of units) in a network. We assume that $\text{nul} \in \mathcal{T}$. A mapping $\pi : K \rightarrow \mathcal{T} \setminus \{\text{nul}\}$ assigns predicates to connections;
- Q is a set of *averaging rules*. A mapping $\alpha : K \rightarrow Q$ determines rules for computing values of connections.

A (surjective) mapping $\mu : \mathcal{U} \rightarrow V$ determines a blockmodel \mathcal{M} of network \mathcal{N} iff it satisfies the conditions:

$$\forall (t, w) \in K : \pi(t, w)(C(t), C(w)) \text{ and}$$

$$\forall (t, w) \in V \times V \setminus K : \text{nul}(C(t), C(w)) .$$



Equivalences

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Let \sim be an equivalence relation over \mathcal{U} and $[X] = \{Y \in \mathcal{U} : X \sim Y\}$. We say that \sim is *compatible* with \mathcal{T} over a network \mathcal{N} iff

$$\forall X, Y \in \mathcal{U} \exists T \in \mathcal{T} : T([X], [Y]).$$

It is easy to verify that the notion of compatibility for $\mathcal{T} = \{\text{nul, reg}\}$ reduces to the usual definition of regular equivalence (White and Reitz 1983). Similarly, compatibility for $\mathcal{T} = \{\text{nul, com}\}$ reduces to structural equivalence (Lorrain and White 1971).

For a compatible equivalence \sim the mapping $\mu: X \mapsto [X]$ determines a blockmodel with $V = \mathcal{U} / \sim$.

The problem of establishing a partition of units in a network in terms of a selected type of equivalence is a special case of **clustering problem** that can be formulated as an optimization problem.



Criterion function

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

One of possible ways of constructing a criterion function that directly reflects the considered equivalence is to measure the fit of a clustering to an ideal one with perfect relations within each cluster and between clusters according to the considered equivalence.

Given a clustering $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$, let $\mathcal{B}(C_u, C_v)$ denote the set of all ideal blocks corresponding to block $R(C_u, C_v)$. Then the global error of clustering \mathbf{C} can be expressed as

$$P(\mathbf{C}) = \sum_{C_u, C_v \in \mathbf{C}} \min_{B \in \mathcal{B}(C_u, C_v)} d(R(C_u, C_v), B)$$

where the term $d(R(C_u, C_v), B)$ measures the difference (error) between the block $R(C_u, C_v)$ and the ideal block B . d is constructed on the basis of characterizations of types of blocks. The function d has to be compatible with the selected type of equivalence.



... criterion functions

For example, for structural equivalence, the term $d(R(C_u, C_v), B)$ can be expressed, for non-diagonal blocks, as

$$d(R(C_u, C_v), B) = \sum_{X \in C_u, Y \in C_v} |r_{XY} - b_{XY}|.$$

where r_{XY} is the observed tie and b_{XY} is the corresponding value in an ideal block. This criterion function counts the number of 1s in erstwhile null blocks and the number of 0s in otherwise complete blocks. These two types of inconsistencies can be weighted differently.

Determining the block error, we also determine the type of the best fitting ideal block (the types are ordered).

The criterion function $P(\mathbf{C})$ is *sensitive* iff $P(\mathbf{C}) = 0 \Leftrightarrow \mu$ (determined by \mathbf{C}) is an exact blockmodeling. For all presented block types sensitive criterion functions can be constructed (Batagelj, 1997).



Deviations Measures for Types of Blocks

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

We can efficiently test whether a block $R(X, Y)$ is of the type T by making use of the characterizations of block types. On this basis we can construct the corresponding deviation measures. The quantities used in the expressions for deviations have the following meaning:

- s_t – total block sum = # of 1s in a block,
- n_r = $\text{card}(X) - \# \text{ of rows in a block}$,
- n_c = $\text{card}(Y) - \# \text{ of columns in a block}$,
- p_r – # of non-null rows in a block,
- p_c – # of non-null columns in a block,
- m_r – maximal row-sum,
- m_c – maximal column-sum,
- s_d – diagonal block sum = # of 1s on a diagonal,
- d – diagonal error = $\min(s_d, n_r - s_d)$.

Throughout the number of elements in a block is $n_r n_c$.



... Deviations Measures for Types of Blocks

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Connection	$\delta(X, Y; T)$	
null	$\begin{cases} s_t \\ s_t + d - s_d \end{cases}$	nondiagonal diagonal
complete	$\begin{cases} n_r n_c - s_t \\ n_r n_c - s_t + d + s_d - n_r \end{cases}$	nondiagonal diagonal
row-dominant	$\begin{cases} (n_c - m_r - 1)n_r \\ (n_c - m_r)n_r \end{cases}$	diagonal, $s_d = 0$ otherwise
col-dominant	$\begin{cases} (n_r - m_c - 1)n_c \\ (n_r - m_c)n_c \end{cases}$	diagonal, $s_d = 0$ otherwise
row-regular	$(n_r - p_r)n_c$	
col-regular	$(n_c - p_c)n_r$	
regular	$(n_c - p_c)n_r + (n_r - p_r)p_c$	
row-functional	$s_t - p_r + (n_r - p_r)n_c$	
col-functional	$s_t - p_c + (n_c - p_c)n_r$	
density γ	$\max(0, \gamma n_r n_c - s_t)$	

For the null, complete, row-dominant and column-dominant blocks it is necessary to distinguish diagonal blocks and non-diagonal blocks.



Solving the blockmodeling problem

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The obtained optimization problem can be solved by local optimization.

Network/Create Partition/Blockmodeling*/

Once a partitioning μ and types of connection π are determined, we can also compute the values of connections by using averaging rules.



Benefits from Optimization Approach

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

- *ordinary / inductive blockmodeling*: Given a network \mathcal{N} and set of types of connection \mathcal{T} , determine the model \mathcal{M} ;
- *evaluation of the quality of a model, comparing different models, analyzing the evolution of a network* (Sampson data, Doreian and Mrvar 1996): Given a network \mathcal{N} , a model \mathcal{M} , and blockmodeling μ , compute the corresponding criterion function;
- *model fitting / deductive blockmodeling*: Given a network \mathcal{N} , set of types \mathcal{T} , and a family of models, determine μ which minimizes the criterion function (Batagelj, Ferligoj, Doreian, 1998).
- we can fit the network to a partial model and analyze the residual afterward;
- we can also introduce different constraints on the model, for example: units X and Y are of the same type; or, types of units X and Y are not connected; ...



Pre-specified blockmodeling

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

In the previous slides the inductive approaches for establishing blockmodels for a set of social relations defined over a set of units were discussed. Some form of equivalence is specified and clusterings are sought that are consistent with a specified equivalence.

Another view of blockmodeling is deductive in the sense of starting with a blockmodel that is specified in terms of substance prior to an analysis.

In this case given a network, set of types of ideal blocks, and a reduced model, a solution (a clustering) can be determined which minimizes the criterion function.



Pre-Specified Blockmodels

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The pre-specified blockmodeling starts with a blockmodel specified, in terms of substance, ***prior to an analysis***. Given a network, a set of ideal blocks is selected, a family of reduced models is formulated, and partitions are established by minimizing the criterion function.

The basic types of models are:

*	*
*	0

center -
periphery

*	0
*	*

hierarchy

*	0
0	*

clustering

0	*
*	0

bipartition



Prespecified blockmodeling example

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

We expect that center-periphery model exists in the network:
some students having good studying material, some not.

Prespecified blockmodel: (com/complete, reg/regular, -/null block)

	1	2
1	[com reg]	-
2	[com reg]	-

Using local optimization we get the partition:

$$\mathbf{C} = \{\{b02, b03, b51, b85, b89, b96, g09\}, \\ \{g07, g10, g12, g22, g24, g28, g42, g63\}\}$$



2 clusters solution

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

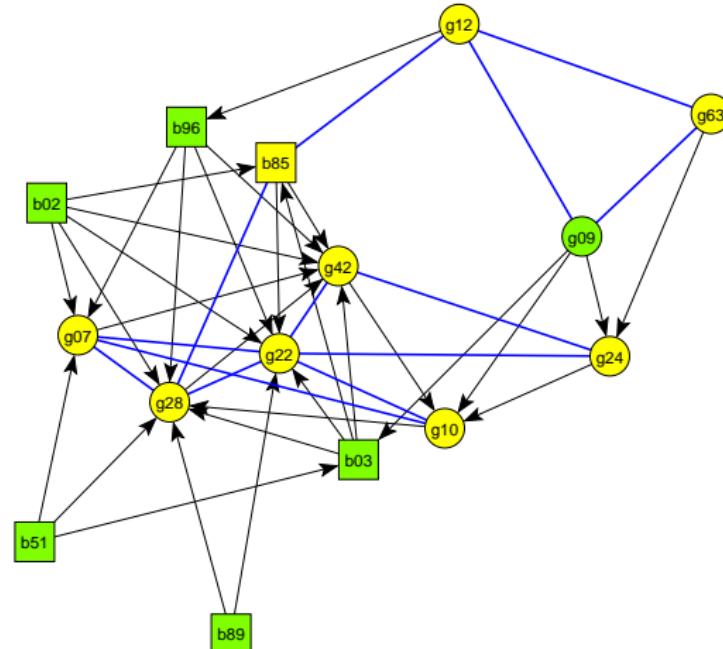
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Model

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

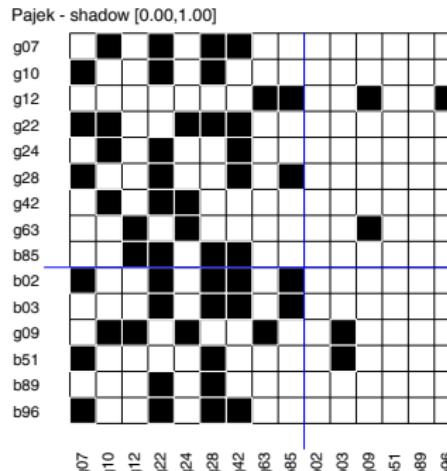


Image and Error Matrices:

	1	2		1	2
1	reg	-	1	0	3
2	reg	-	2	0	2

Total error = 5

Operations/Network+Partition/Blockmodeling*/Random Start



The Student Government at the University of Ljubljana in 1992

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

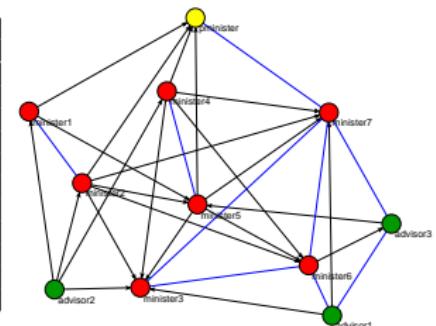
Generalizations

Clustering
with
constraints

The relation is determined by the following question (Hlebec, 1993):

*Of the members and advisors of the Student Government,
whom do you most often talk with about the matters of the Student Government?*

		m 1	p 2	m 3	m 4	m 5	m 6	m 7	m 8	a 9	a 10	a 11
minister 1	1	.	1	1	.	.	1
p.minister	2	1	.	.	.
minister 2	3	1	1	.	1	.	1	1	1	.	.	.
minister 3	4	1	1	.	.	.
minister 4	5	.	1	.	1	1	.	1	1	.	.	.
minister 5	6	.	1	.	1	1	.	1	1	.	.	.
minister 6	7	.	.	.	1	.	.	.	1	1	.	1
minister 7	8	.	1	.	1	1	.	1	.	.	.	1
adviser 1	9	.	.	.	1	.	.	1	1	.	.	1
adviser 2	10	1	.	1	1	1
adviser 3	11	1	.	1	1	.	.





A Symmetric Acyclic Blockmodel of Student Government

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

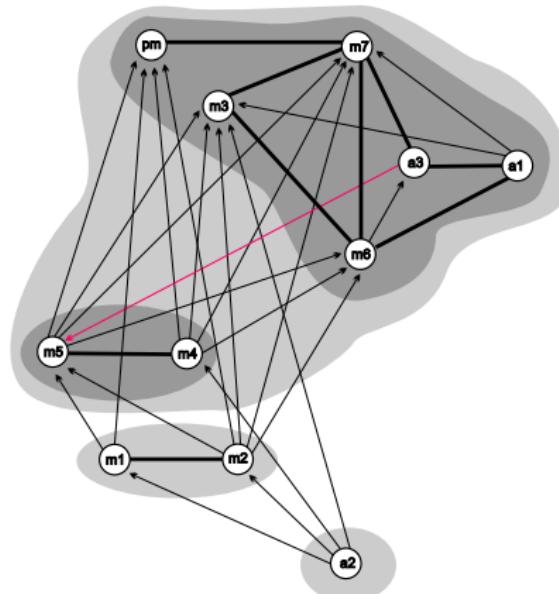
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



The obtained clustering in 4 clusters is almost exact. The only error is produced by the arc $(a3, m5)$.

Network/Create Hierarchy/Symmetric-Acyclic



Football

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

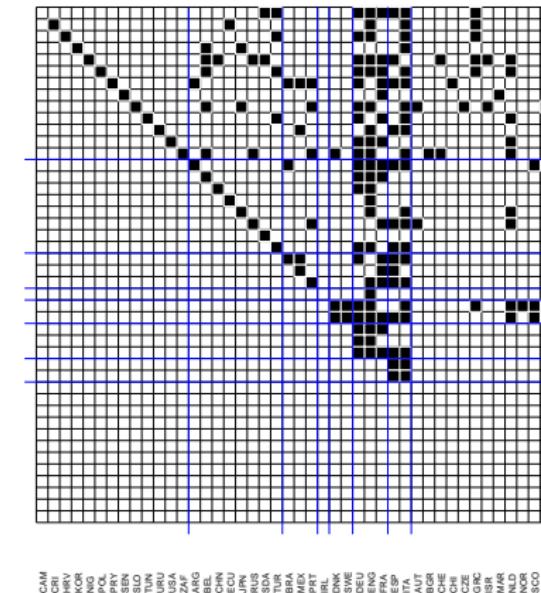
Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

Pajek - shadow [0.00,1.00]



The player's market of the Fifa Football Worldchampionship 2002 (Japan/Korea). The data, collected by L. Krempel, describe the 733 players of all 32 participating national teams and the clubs and countries where each of these players have contracts.

For acyclic (below diagonal blocks are zero-blocks) regular blockmodel we get a solution with 8 clusters and Error = 30



Demo with Pajek

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

```
Read Network [Tina.net]
Network/Create New Network/Transform/Arcs->Edges/
    Bidirected Only/Max
Draw/Network
Layout/Energy/Kamada-Kawai/Free
Operations/Network+Partition/Blockmodeling*/
    Restricted Options [On]
Operations/Network+Partition/Blockmodeling*/Random
Start
    [4, Ranks.MDL], [Repetitions, 100], [Clusters, 4],
[RUN]
    extend the dialog box to see the model
Draw/Network+FirstPartition
```



Blockmodeling in 2-mode networks

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

We already presented some ways of rearranging 2-mode network matrices at the beginning of this lecture.

It is also possible to formulate this goal as a generalized blockmodeling problem where the solutions consist of two partitions — row-partition and column-partition.



Supreme Court Voting for Twenty-Six Important Decisions

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

Issue	Label	Br	Gi	So	St	OC	Ke	Re	Sc	Th
Presidential Election	PE	-	-	-	-	+	+	+	+	+
Criminal Law Cases										
Illegal Search 1	CL1	+	+	+	+	+	+	-	-	-
Illegal Search 2	CL2	+	+	+	+	+	+	-	-	-
Illegal Search 3	CL3	+	+	+	-	-	-	-	+	+
Seat Belts	CL4	-	-	+	-	-	+	+	+	+
Stay of Execution	CL5	+	+	+	+	+	+	-	-	-
Federal Authority Cases										
Federalism	FA1	-	-	-	-	+	+	+	+	+
Clean Air Action	FA2	+	+	+	+	+	+	+	+	+
Clean Water	FA3	-	-	-	-	+	+	+	+	+
Cannabis for Health	FA4	0	+	+	+	+	+	+	+	+
United Foods	FA5	-	-	+	+	-	+	+	+	+
NY Times Copyrights	FA6	-	+	+	-	+	+	+	+	+
Civil Rights Cases										
Voting Rights	CR1	+	+	+	+	+	-	-	-	-
Title VI Disabilities	CR2	-	-	-	-	+	+	+	+	+
PGA v. Handicapped Player	CR3	+	+	+	+	+	+	+	-	-
Immigration Law Cases										
Immigration Jurisdiction	Im1	+	+	+	+	-	+	-	-	-
Deporting Criminal Aliens	Im2	+	+	+	+	+	-	-	-	-
Detaining Criminal Aliens	Im3	+	+	+	+	-	+	-	-	-
Citizenship	Im4	-	-	-	+	-	+	+	+	+
Speech and Press Cases										
Legal Aid for Poor	SP1	+	+	+	+	-	+	-	-	-
Privacy	SP2	+	+	+	+	+	+	-	-	-
Free Speech	SP3	+	-	-	-	+	+	+	+	+
Campaign Finance	SP4	+	+	+	+	+	-	-	-	-
Tobacco Ads	SP5	-	-	-	+	+	+	+	+	+
Labor and Property Rights Cases										
Labor Rights	LPR1	-	-	-	-	+	+	+	+	+
Property Rights	LPR2	-	-	-	-	+	+	+	+	+

The Supreme Court Justices and their 'votes' on a set of 26 "important decisions" made during the 2000-2001 term, Doreian and Fujimoto (2002).

The Justices (in the order in which they joined the Supreme Court) are:

Rehnquist (1972), Stevens (1975), O'Connor (1981), Scalia (1982), Kennedy (1988), Souter (1990), Ginsburg (1993) and Breyer (1994).



... Supreme Court Voting / a (4,7) partition

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

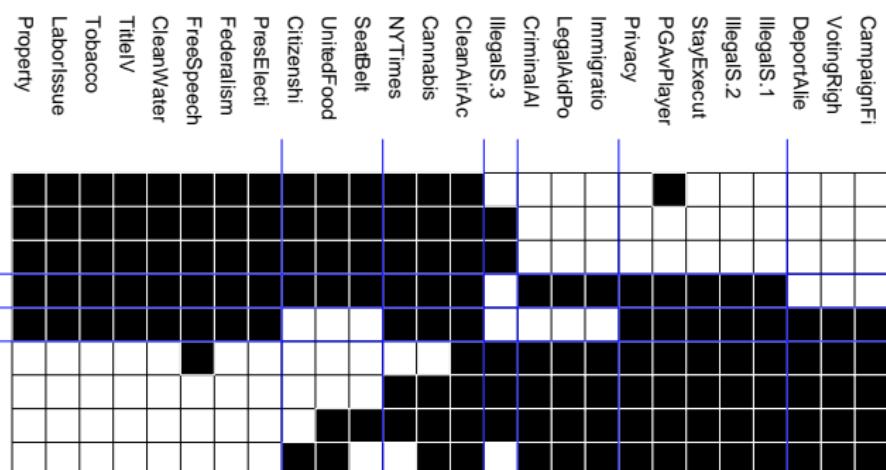
Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints



upper – conservative / lower – liberal



Signed graphs

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

A *signed graph* is an ordered pair (\mathcal{G}, σ) where

- $\mathcal{G} = (\mathcal{V}, R)$ is a directed graph (without loops) with set of nodes \mathcal{V} and set of arcs $R \subseteq \mathcal{V} \times \mathcal{V}$;
- $\sigma : R \rightarrow \{p, n\}$ is a *sign* function. The arcs with the sign p are *positive* and the arcs with the sign n are *negative*. We denote the set of all positive arcs by R^+ and the set of all negative arcs by R^- .

The case when the graph is undirected can be reduced to the case of directed graph by replacing each edge e by a pair of opposite arcs both signed with the sign of the edge e .



Balanced and clusterable signed graphs

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The signed graphs were introduced in Harary, 1953 and later studied by several authors. Following Roberts (1976, p. 75–77) a signed graph (\mathcal{G}, σ) is:

- *balanced* iff the set of nodes \mathcal{V} can be partitioned into two subsets so that every positive arc joins nodes of the same subset and every negative arc joins nodes of different subsets.
- *clusterable* iff the set of \mathcal{V} can be partitioned into subsets, called *clusters*, so that every positive arc joins nodes of the same subset and every negative arc joins nodes of different subsets.



... Properties

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The (semi)walk on the signed graph is *positive* iff it contains an even number of negative arcs; otherwise it is *negative*.

The balanced and clusterable signed graphs are characterised by the following theorems:

THEOREM 1. A signed graph (\mathcal{G}, σ) is balanced iff every closed semiwalk is positive.

THEOREM 2. A signed graph (\mathcal{G}, σ) is clusterable iff \mathcal{G} contains no closed semiwalk with exactly one negative arc.



Chartrand's example – graph

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

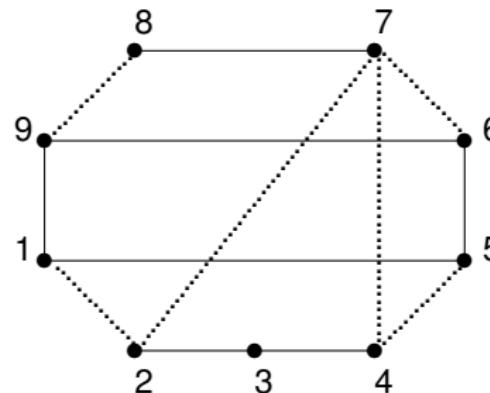
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



	1	2	3	4	5	6	7	8	9
1	0	<i>n</i>	0	0	<i>p</i>	0	0	0	<i>p</i>
2	<i>n</i>	0	<i>p</i>	0	0	0	<i>n</i>	0	0
3	0	<i>p</i>	0	<i>p</i>	0	0	0	0	0
4	0	0	<i>p</i>	0	<i>n</i>	0	<i>n</i>	0	0
5	<i>p</i>	0	0	<i>n</i>	0	<i>p</i>	0	0	0
6	0	0	0	0	<i>p</i>	0	<i>n</i>	0	<i>p</i>
7	0	<i>n</i>	0	<i>n</i>	0	<i>n</i>	0	<i>p</i>	0
8	0	0	0	0	0	0	<i>p</i>	0	<i>n</i>
9	<i>p</i>	0	0	0	0	<i>p</i>	0	<i>n</i>	0

In the figure the graph from Chartrand (1985, p. 181) and its value matrix are given. The positive edges are drawn with solid lines, and the negative edges with dotted lines.



Chartrand's example – closures

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

	1	2	3	4	5	6	7	8	9
1	a	a	a	a	a	a	a	a	a
2	a	a	a	a	a	a	a	a	a
3	a	a	a	a	a	a	a	a	a
4	a	a	a	a	a	a	a	a	a
5	a	a	a	a	a	a	a	a	a
6	a	a	a	a	a	a	a	a	a
7	a	a	a	a	a	a	a	a	a
8	a	a	a	a	a	a	a	a	a
9	a	a	a	a	a	a	a	a	a
	1	2	3	4	5	6	7	8	9
1	p	n	n	n	p	p	n	n	p
2	n	p	p	p	n	n	n	n	n
3	n	p	p	p	n	n	n	n	n
4	n	p	p	p	n	n	n	n	n
5	p	n	n	n	p	p	n	n	p
6	p	n	n	n	p	p	n	n	p
7	n	n	n	n	n	n	p	p	n
8	n	n	n	n	n	n	p	p	n
9	p	n	n	n	p	p	n	n	p

On the left side of the table the corresponding balance-closure is given – the graph is not balanced. From the cluster-closure on the right side of the table we can see that the graph is clusterable and it has the clusters

$$\mathcal{V}_1 = \{1, 5, 6, 9\}, \quad \mathcal{V}_2 = \{2, 3, 4\}, \quad \mathcal{V}_3 = \{7, 8\}$$



Clusterability and blockmodeling

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

To the sign graph clusterability problem correspond three types of blocks:

- *null* all elements in a block are 0;
- *positive* all elements in a block are positive or 0;
- *negative* all elements in a block are negative or 0;

If a graph is clusterable the blocks determined by the partition are: positive or null on the diagonal; and negative or null outside the diagonal.

The clusterability of partition $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$ can be therefore measured as follows ($0 \leq \alpha \leq 1$):

$$P_\alpha(\mathbf{C}) = \alpha \sum_{C \in \mathbf{C}} \sum_{u, v \in C} \max(0, -w_{uv}) + (1-\alpha) \sum_{\substack{c, c' \in \mathbf{C} \\ c \neq c'}} \sum_{u \in C, v \in C'} \max(0, w_{uv})$$

The blockmodeling problem can be solved by local optimization.



Slovenian political parties 1994 (S. Kropivnik)

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

		1	2	3	4	5	6	7	8	9	10
	SKD	1	0	-215	114	-89	-77	94	-170	176	117
	ZLSD	2	-215	0	-217	134	77	-150	57	-253	-230
	SDSS	3	114	-217	0	-203	-80	138	-109	177	180
	LDS	4	-89	134	-203	0	157	-142	173	-241	-254
	ZSESS	5	-77	77	-80	157	0	-188	170	-120	-160
	ZS	6	94	-150	138	-142	-188	0	-97	140	116
	DS	7	-170	57	-109	173	170	-97	0	-184	-191
	SLS	8	176	-253	177	-241	-120	140	-184	0	235
	SPS-SNS	9	117	-230	180	-254	-160	116	-191	235	0
	SNS	10	-210	49	-174	23	-9	-106	-6	-132	-164

SKD – Slovene Christian Democrats; ZLSD – Associated List of Social Democrats;

SDSS – Social Democratic Party of Slovenia; LDS – Liberal Democratic Party;

ZSESS and ZS – two Green Parties, separated after 1992 elections; DS – Democratic Party;

SLS – Slovene People's Party; SNS – Slovene National Party;

SPS SNS – a group of deputies, former members of SNS, separated after 1992 elections

Network **Stranke94**.



Slovenian political parties 1994 / reordered

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

		1	3	6	8	9	2	4	5	7	10
SKD	1	0	114	94	176	117	-215	-89	-77	-170	-210
SDSS	3	114	0	138	177	180	-217	-203	-80	-109	-174
ZS	6	94	138	0	140	116	-150	-142	-188	-97	-106
SLS	8	176	177	140	0	235	-253	-241	-120	-184	-132
SPS-SNS	9	117	180	116	235	0	-230	-254	-160	-191	-164
ZLSD	2	-215	-217	-150	-253	-230	0	134	77	57	49
LDS	4	-89	-203	-142	-241	-254	134	0	157	173	23
ZSESS	5	-77	-80	-188	-120	-160	77	157	0	170	-9
DS	7	-170	-109	-97	-184	-191	57	173	170	0	-6
SNS	10	-210	-174	-106	-132	-164	49	23	-9	-6	0

S. Kropivnik, A. Mrvar: An Analysis of the Slovene Parliamentary Parties Network. in Developments in data analysis, MZ 12, FDV, Ljubljana, 1996, p. 209-216.



3-way blockmodeling

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

We started to work on blockmodeling of 3-way networks. We developed the indirect approach to *structural equivalence blockmodeling in 3-way networks*. *Indirect* means – embedding the notion of equivalence in a dissimilarity and determining it using clustering.

3-way network is defined by three sets of units X , Y and Z . There are three basic cases:

- all three sets are different (3-mode netork)
- two sets are the same (2-mode network)
- all three sets are the same (1-mode network)

For all three cases we constructed compatible dissimilarities for structural equivalence .



Example 1: Artificial dataset

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

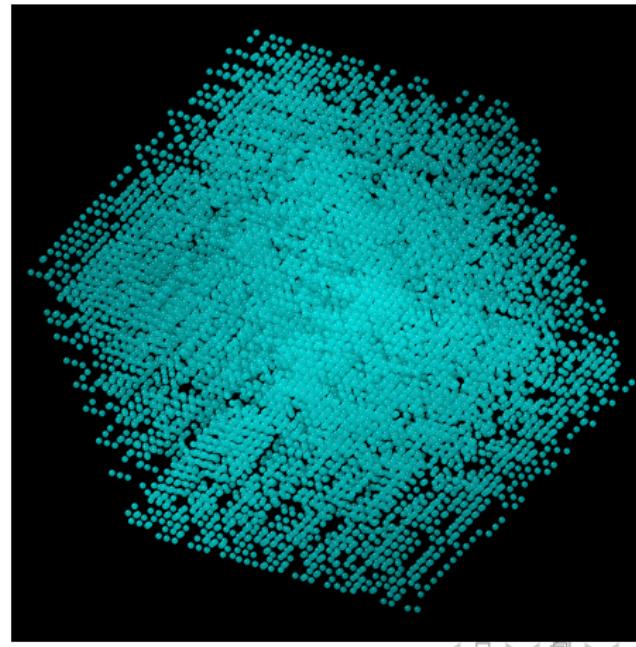
Signed graphs

Generalizations

Clustering
with
constraints

Randomly generated ideal structure

rndTest (c(5, 6, 4), c(35, 35, 35)):





Example 1: Dendrograms

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

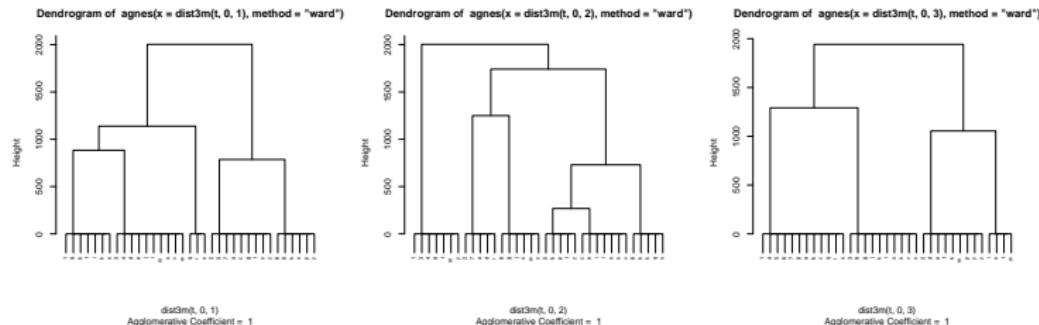
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Example 1: Solutions

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

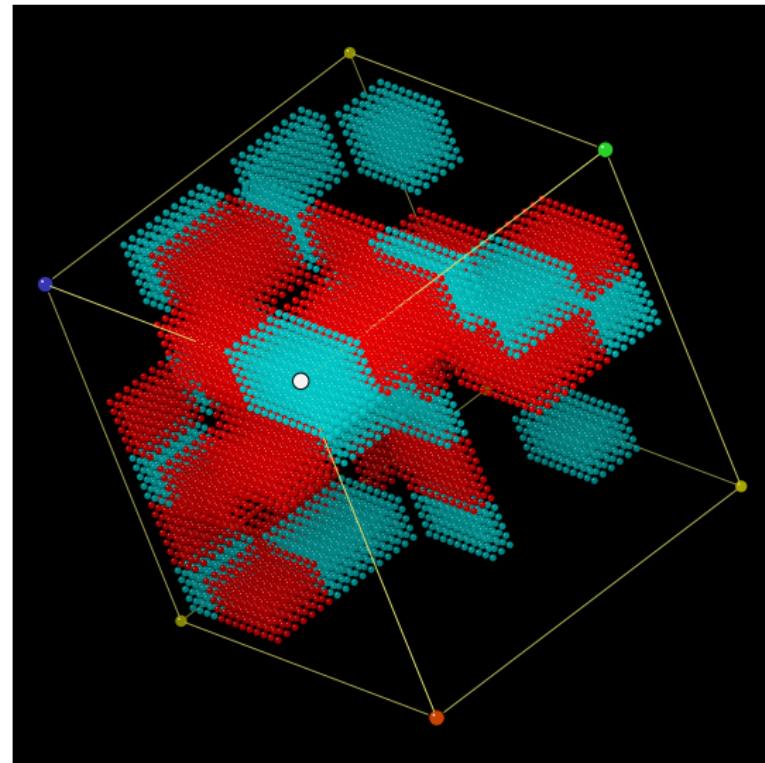
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Example 2: Krackhardt / Dendrograms

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

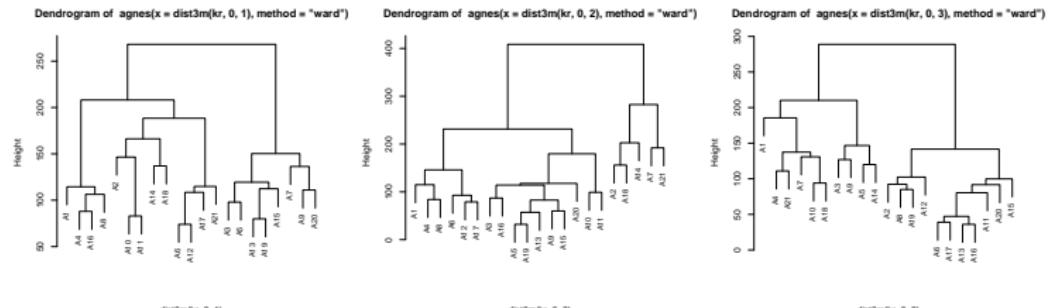
Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints





Example 2: Krackhardt / Solutions

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

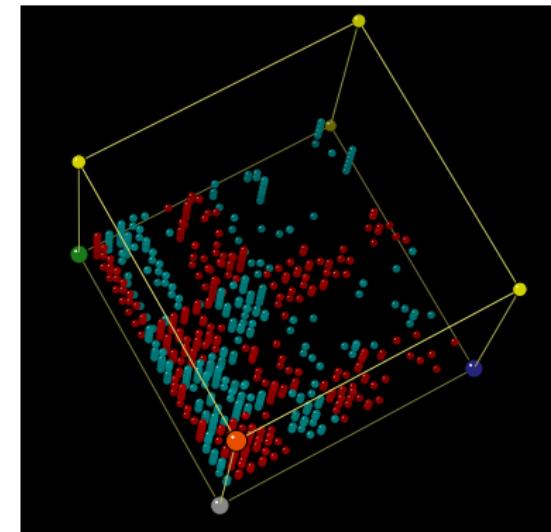
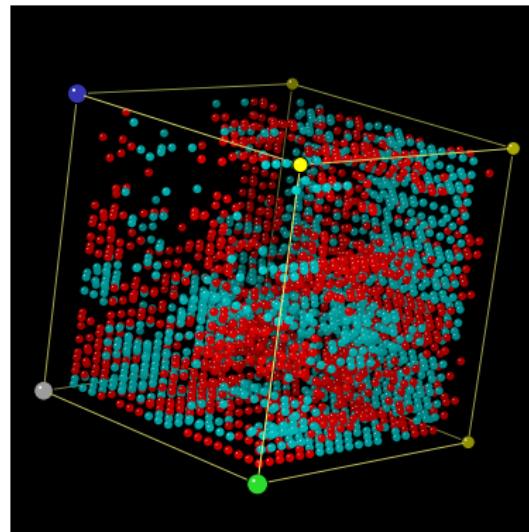
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Blockmodeling of Valued Networks

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Batagelj and Ferligoj (2000) proposed an approach to blockmodel of valued networks as an example of *relational data analysis*. These ideas were further developed by Žiberna (2007) who proposed some approaches for generalized blockmodeling of valued networks.

The first one is a straightforward generalization of the generalized blockmodeling of binary networks to valued blockmodeling. The second approach is homogeneity blockmodeling where the basic idea is that the inconsistency of an empirical block with its ideal block can be measured by within block variability of appropriate values. Žiberna provided new ideal blocks appropriate for blockmodeling of valued networks together with definitions of their block inconsistencies.



More on blockmodeling

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

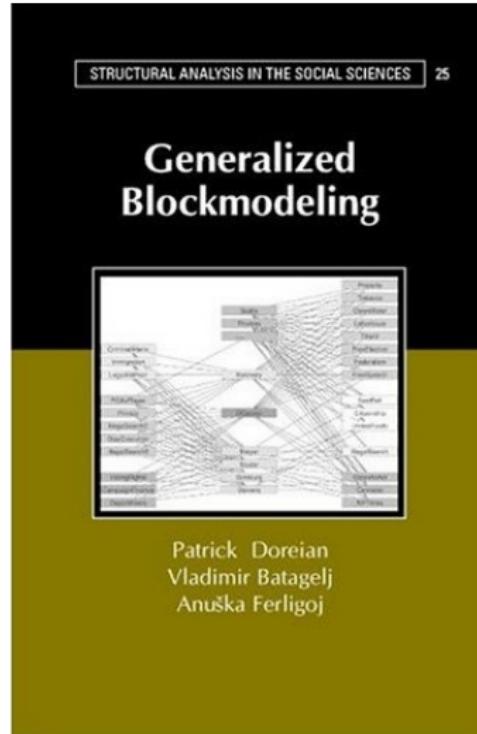
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



The details about the generalized blockmodeling can be found in our book:

P. Doreian, V. Batagelj, A. Ferligoj: *Generalized Blockmod-
eling*, CUP, 2005.



Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

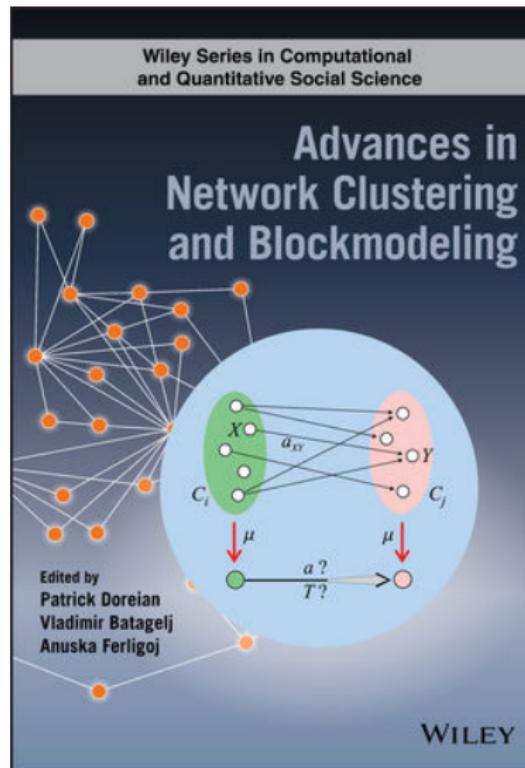
Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

... More on blockmodeling



Wiley 2020
Amazon



Final Remarks

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The current, local optimization based, programs for generalized blockmodeling can deal only with networks with at most some hundreds of units. What to do with larger networks is an open question. For some specialized problems also procedures for (very) large networks can be developed (Doreian, Batagelj, Ferligoj, 1998; Batagelj, Zaveršnik, 2002).

Another interesting problem is the development of *blockmodeling of valued networks* or more general *relational data analysis* (Batagelj, Ferligoj, 2000).



Conditions for hierarchical clustering methods

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The set of feasible clusterings Φ determines the *feasibility predicate* $\Phi(\mathbf{C}) \equiv \mathbf{C} \in \Phi$ defined on $\mathcal{P}(\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\})$; and conversely $\Phi \equiv \{\mathbf{C} \in \mathcal{P}(\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}) : \Phi(\mathbf{C})\}$.

In the set Φ the relation of *clustering inclusion* \sqsubseteq can be introduced by

$$\mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \equiv \forall C_1 \in \mathbf{C}_1, C_2 \in \mathbf{C}_2 : C_1 \cap C_2 \in \{\emptyset, C_1\}$$

we say also that the clustering \mathbf{C}_1 is a *refinement* of the clustering \mathbf{C}_2 .

It is well known that $(\Pi(\mathcal{U}), \sqsubseteq)$ is a partially ordered set (even more, semimodular lattice). Because any subset of partially ordered set is also partially ordered, we have: Let $\Phi \subseteq \Pi(\mathcal{U})$ then (Φ, \sqsubseteq) is a partially ordered set.

The clustering inclusion determines two related relations (on Φ):

$\mathbf{C}_1 \sqsubset \mathbf{C}_2 \equiv \mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \wedge \mathbf{C}_1 \neq \mathbf{C}_2$ – strict inclusion, and

$\mathbf{C}_1 \sqleftarrow \mathbf{C}_2 \equiv \mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \wedge \neg \exists \mathbf{C} \in \Phi : (\mathbf{C}_1 \sqsubset \mathbf{C} \wedge \mathbf{C} \sqsubset \mathbf{C}_2)$ – predecessor.



Conditions on the structure of the set of feasible clusterings

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

We shall assume that the set of feasible clusterings $\Phi \subseteq \Pi(\mathcal{U})$ satisfies the following conditions:

F1. $\mathbf{O} \equiv \{\{X\} : X \in \mathcal{U}\} \in \Phi$

F2. The feasibility predicate Φ is *local* – it has the form $\Phi(\mathbf{C}) = \bigwedge_{C \in \mathbf{C}} \varphi(C)$ where $\varphi(C)$ is a predicate defined on $\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$ (clusters).

The intuitive meaning of $\varphi(C)$ is: $\varphi(C) \equiv$ the cluster C is 'good'. Therefore the locality condition can be read: a 'good' clustering $\mathbf{C} \in \Phi$ consists of 'good' clusters.

F3. The predicate Φ has the property of *binary heredity* with respect to the *fusibility* predicate $\psi(C_1, C_2)$, i.e.,

$$C_1 \cap C_2 = \emptyset \wedge \varphi(C_1) \wedge \varphi(C_2) \wedge \psi(C_1, C_2) \Rightarrow \varphi(C_1 \cup C_2)$$

This condition means: in a 'good' clustering, a fusion of two 'fusible' clusters produces a 'good' clustering.



... conditions

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

F4. The predicate ψ is *compatible* with clustering inclusion \sqsubseteq , i.e.,

$$\forall \mathbf{C}_1, \mathbf{C}_2 \in \Phi : (\mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \wedge \mathbf{C}_1 \setminus \mathbf{C}_2 = \{C_1, C_2\} \Rightarrow \psi(C_1, C_2) \vee \psi(C_2, C_1))$$

F5. The *interpolation* property holds in Φ , i.e., $\forall \mathbf{C}_1, \mathbf{C}_2 \in \Phi :$

$$(\mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \wedge \text{card}((\mathbf{C}_1)) > \text{card}((\mathbf{C}_2)) + 1 \Rightarrow \exists \mathbf{C} \in \Phi : (\mathbf{C}_1 \sqsubseteq \mathbf{C} \wedge \mathbf{C} \sqsubseteq \mathbf{C}_2))$$

These conditions provide a framework in which the hierarchical methods can be applied also for constrained clustering problems $\Phi_k(\mathcal{U}) \subset \Pi_k(\mathcal{U})$.

In the ordinary problem both predicates $\varphi(C)$ and $\psi(C_p, C_q)$ are always true – all conditions F1-F5 are satisfied.



Clustering with relational constraint

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Suppose that the units are described by attribute data $a: \mathcal{U} \rightarrow [\mathcal{U}]$ and related by a binary *relation* $R \subseteq \mathcal{U} \times \mathcal{U}$ that determine the *relational data* (\mathcal{U}, R, a) .

We want to cluster the units according to the similarity of their descriptions, but also considering the relation R – it imposes *constraints* on the set of feasible clusterings, usually in the following form:

$$\Phi(R) = \{\mathbf{C} \in P(\mathcal{U}) : \text{each cluster } C \in \mathbf{C} \text{ is a subgraph } (C, R \cap C \times C) \text{ in the graph } (\mathcal{U}, R) \text{ of the required type of connectedness}\}$$

Example: regionalization problem – group given territorial units into regions such that units inside the region will be similar and form contiguous part of the territory.



Some types of relational constraints

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

We can define different types of sets of feasible clusterings for the same relation R . Some examples of *types of relational constraint* $\Phi^i(R)$ are

type	type of connectedness
$\Phi^1(R)$	weakly connected units
$\Phi^2(R)$	weakly connected units that contain at most one center
$\Phi^3(R)$	strongly connected units
$\Phi^4(R)$	clique
$\Phi^5(R)$	the existence of a trail containing all the units of the cluster

Trail – all arcs are distinct.

A set of units $L \subseteq C$ is a *center* of cluster C in the clustering of type $\Phi^2(R)$ iff the subgraph induced by L is strongly connected and $R(L) \cap (C \setminus L) = \emptyset$.



Some graphs of different types

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

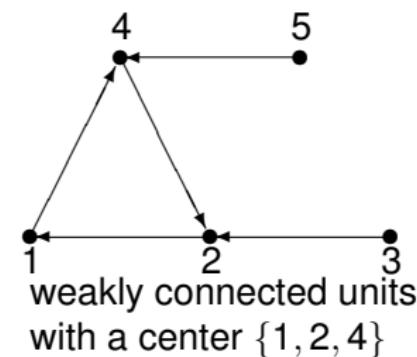
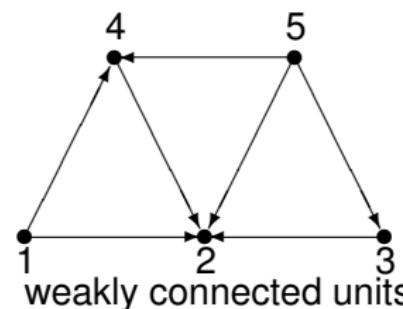
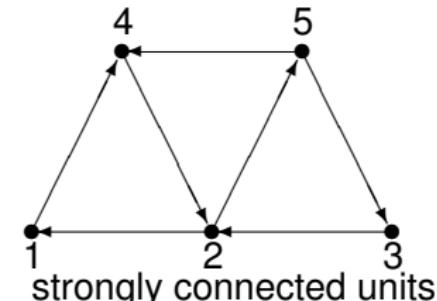
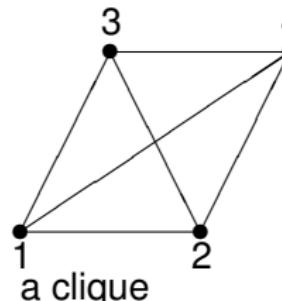
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Properties of relational constraints

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The sets of feasible clusterings $\Phi^i(R)$ are linked as follows:

$$\Phi^4(R) \subseteq \Phi^3(R) \subseteq \Phi^2(R) \subseteq \Phi^1(R)$$

$$\Phi^4(R) \subseteq \Phi^5(R) \subseteq \Phi^2(R)$$

If the relation R is symmetric, then $\Phi^3(R) = \Phi^1(R)$

If the relation R is an equivalence relation, then $\Phi^4(R) = \Phi^1(R)$

Here are also examples of the corresponding fusibility predicates:

$$\psi^1(C_1, C_2) \equiv \exists X \in C_1 \exists Y \in C_2 : (XRY \vee YRX)$$

$$\psi^2(C_1, C_2) \equiv (\exists X \in L_1 \exists Y \in C_2 : XRY) \vee (\exists X \in C_1 \exists Y \in L_2 : YRX)$$

$$\psi^3(C_1, C_2) \equiv (\exists X \in C_1 \exists Y \in C_2 : XRY) \wedge (\exists X \in C_1 \exists Y \in C_2 : YRX)$$

$$\psi^4(C_1, C_2) \equiv \forall X \in C_1 \forall Y \in C_2 : (XRY \wedge YRX)$$

$$\psi^5(C_1, C_2) \equiv (\exists X \in T_1 \exists Y \in I_2 : XRY) \vee (\exists X \in I_1 \exists Y \in T_2 : YRX)$$

For ψ^3 the property F5 fails.



Agglomerative method for relational constraints

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

We can use both hierarchical and local optimization methods for solving some types of problems with relational constraint (Ferligoj, Batagelj 1983).

1. $k := n; \mathbf{C}(k) := \{\{X\} : X \in \mathcal{U}\};$
2. **while** $\exists C_i, C_j \in \mathbf{C}(k): (i \neq j \wedge \psi(C_i, C_j))$ **repeat**
 - 2.1. $(C_p, C_q) := \operatorname{argmin}\{D(C_i, C_j) : i \neq j \wedge \psi(C_i, C_j)\};$
 - 2.2. $C := C_p \cup C_q; k := k - 1;$
 - 2.3. $\mathbf{C}(k) := \mathbf{C}(k + 1) \setminus \{C_p, C_q\} \cup \{C\};$
 - 2.4. determine $D(C, C_s)$ for all $C_s \in \mathbf{C}(k)$
 - 2.4. adjust the relation R as required by the clustering type
3. $m := k$

The condition $\psi(C_i, C_j)$ is equivalent to $C_i RC_j$ for tolerant, leader and strict method; and to $C_i RC_j \wedge C_j RC_i$ for two-way method.



Adjusting relation after joining

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

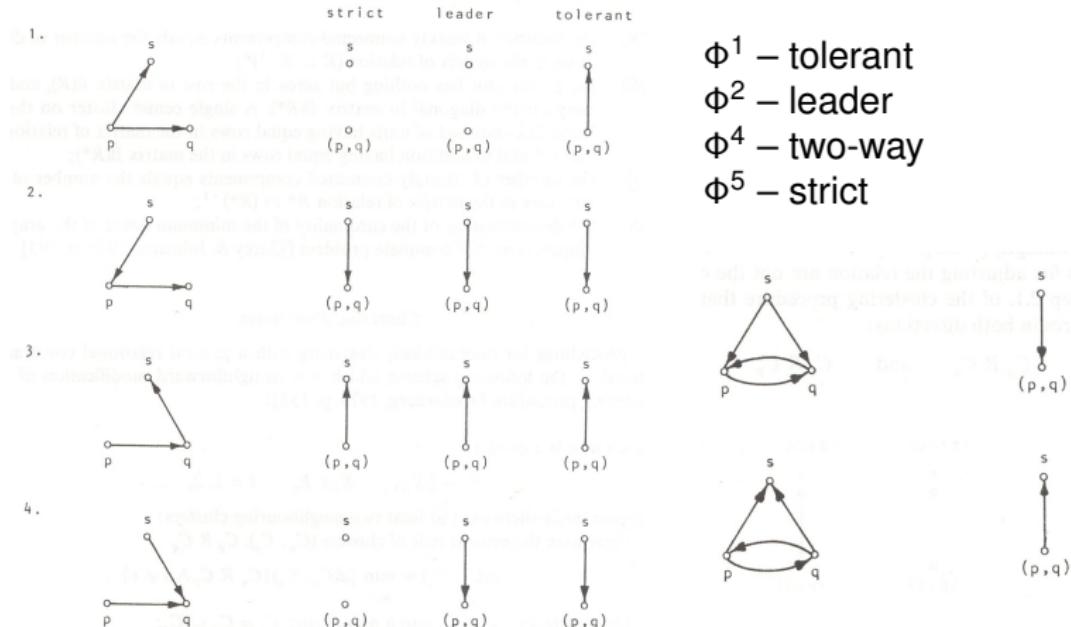
Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints





Example - problem

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

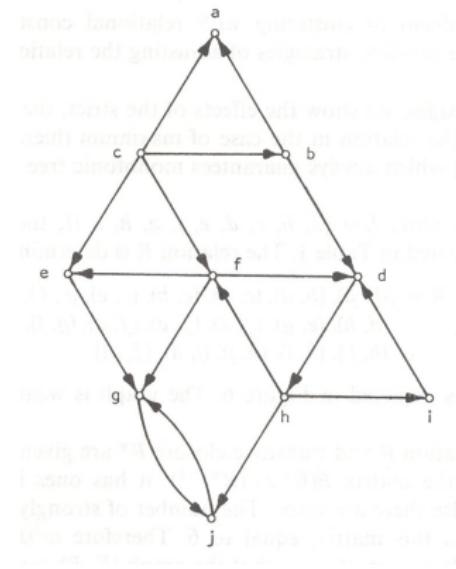
Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

	a	b	c	d	e	f	g	h	i	j
a	0	5	7	4	6	6	2	4	2	3
b		0	8	1	3	4	4	5	3	4
c			0	4	5	7	9	3	2	5
d				0	3	2	4	8	6	3
e					0	6	4	6	5	7
f						0	6	8	5	7
g							0	4	8	2
h								0	3	4
i									0	5
j										0





Example - solution

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

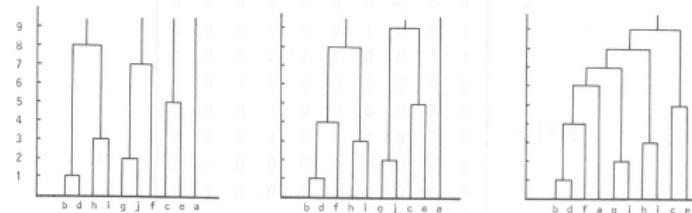
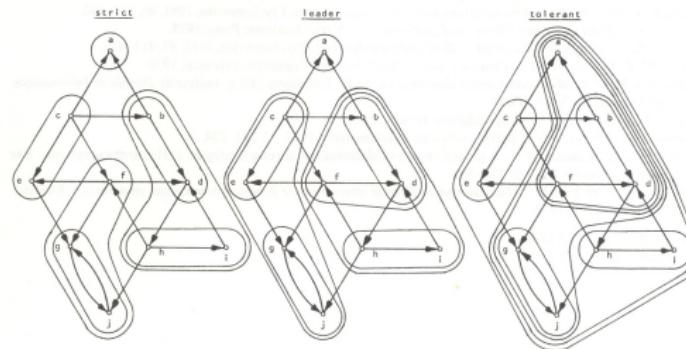
Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints





Dissimilarities between clusters

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

In the original approach a complete dissimilarity matrix is needed.
To obtain fast algorithms we propose to *consider only the dissimilarities between linked units.*

Let (\mathcal{U}, R) , $R \subseteq \mathcal{U} \times \mathcal{U}$ be a graph and $\emptyset \subset S, T \subset \mathcal{U}$ and $S \cap T = \emptyset$.

We call a *block* of relation R for S and T its part
 $R(S, T) = R \cap S \times T$.

The *symmetric closure* of relation R we denote with $\hat{R} = R \cup R^{-1}$.
It holds: $\hat{R}(S, T) = \hat{R}(T, S)$.

For all dissimilarities between clusters $D(S, T)$ we set:

$$D(\{s\}, \{t\}) = \begin{cases} d(s, t) & s \hat{R} t \\ \infty & \text{otherwise} \end{cases}$$

where d is a selected dissimilarity between units.



Minimum and Maximum

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Minimum

$$D_{\min}(S, T) = \min_{(s,t) \in \hat{R}(S,T)} d(s, t)$$

$$D_{\min}(S, T_1 \cup T_2) = \min(D_{\min}(S, T_1), D_{\min}(S, T_2))$$

Maximum

$$D_{\max}(S, T) = \max_{(s,t) \in \hat{R}(S,T)} d(s, t)$$

$$D_{\max}(S, T_1 \cup T_2) = \max(D_{\max}(S, T_1), D_{\max}(S, T_2))$$



Average

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

$w : V \rightarrow \mathbb{R}$ – is a weight on units; for example $w(v) = 1$, for all $v \in \mathcal{U}$.

$$D_a(S, T) = \frac{1}{w(\hat{R}(S, T))} \sum_{(s, t) \in \hat{R}(S, T)} d(s, t)$$

$$w(\hat{R}(S, T_1 \cup T_2)) = w(\hat{R}(S, T_1)) + w(\hat{R}(S, T_2))$$

$$D_a(S, T_1 \cup T_2) = \frac{w(\hat{R}(S, T_1))}{w(\hat{R}(S, T_1 \cup T_2))} D_a(S, T_1) + \frac{w(\hat{R}(S, T_2))}{w(\hat{R}(S, T_1 \cup T_2))} D_a(S, T_2)$$



Hierarchies

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The agglomerative clustering procedure produces a series of feasible clusterings $\mathbf{C}(n), \mathbf{C}(n-1), \dots, \mathbf{C}(m)$ with $\mathbf{C}(m) \in \text{Max } \Phi$ (maximal elements for \sqsubseteq).

Their union $\mathcal{T} = \bigcup_{k=m}^n \mathbf{C}(k)$ is called a *hierarchy* and has the property

$$\forall C_p, C_q \in \mathcal{T} : C_p \cap C_q \in \{\emptyset, C_p, C_q\}$$

The set inclusion \subseteq is a *tree* or *hierarchical* order on \mathcal{T} . The hierarchy \mathcal{T} is *complete* iff $\mathcal{U} \in \mathcal{T}$.

For $W \subseteq \mathcal{U}$ we define the *smallest cluster* $C_{\mathcal{T}}(W)$ from \mathcal{T} containing W as:

c1. $W \subseteq C_{\mathcal{T}}(W)$

c2. $\forall C \in \mathcal{T} : (W \subseteq C \Rightarrow C_{\mathcal{T}}(W) \subseteq C)$

$C_{\mathcal{T}}$ is a *closure* on \mathcal{T} with a special property

$$Z \notin C_{\mathcal{T}}(\{X, Y\}) \Rightarrow C_{\mathcal{T}}(\{X, Y\}) \subset C_{\mathcal{T}}(\{X, Y, Z\}) = C_{\mathcal{T}}(\{X, Z\}) = C_{\mathcal{T}}(\{Y, Z\})$$



Level functions

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

A mapping $h : \mathcal{T} \rightarrow \mathbb{R}_0^+$ is a *level function* on \mathcal{T} iff

I1. $\forall X \in \mathcal{U} : h(\{X\}) = 0$

I2. $C_p \subseteq C_q \Rightarrow h(C_p) \leq h(C_q)$

A simple example of level function is $h(C) = \text{card}((C)) - 1$.

Every hierarchy / level function determines an ultrametric dissimilarity on \mathcal{U}

$$\delta(X, Y) = h(C_{\mathcal{T}}(\{X, Y\}))$$

The converse is also true (see Dieudonne (1960)): Let d be an ultrametric on \mathcal{U} . Denote $\overline{B}(X, r) = \{Y \in \mathcal{U} : d(X, Y) \leq r\}$. Then for any given set $A \subset \mathbb{R}^+$ the set

$$\mathbf{C}(A) = \{\overline{B}(X, r) : X \in \mathcal{U}, r \in A\} \cup \{\{\mathcal{U}\}\} \cup \{\{X\} : X \in \mathcal{U}\}$$

is a complete hierarchy, and $h(C) = \text{diam}(C)$ is a level function.

The pair (\mathcal{T}, h) is called a *dendrogram* or a *clustering tree* because it can be visualized as a tree.



Reducibility

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The dissimilarity D has the *reducibility* property (Bruynooghe, 1977) iff

$$D(C_p, C_q) \leq \min(D(C_p, C_s), D(C_q, C_s)) \Rightarrow$$

$$\min(D(C_p, C_s), D(C_q, C_s)) \leq D(C_p \cup C_q, C_s)$$

or equivalently

$$D(C_p, C_q) \leq t, D(C_p, C_s) \geq t, D(C_q, C_s) \geq t \Rightarrow D(C_p \cup C_q, C_s) \geq t$$

Theorem

If a dissimilarity D has the reducibility property then h_D is a level function.



Nearest neighbors graphs

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

For a given dissimilarity d on the set of units \mathcal{U} and relational constraint R we define the *k nearest neighbors graph*

$$\mathbf{G}_{NN} = (\mathcal{U}, A)$$

$(X, Y) \in A \Leftrightarrow Y$ is selected among the nearest neighbors of X and $X R Y$

By setting for $(X, Y) \in A$ its value to $w((X, Y)) = d(X, Y)$ we obtain a network $\mathcal{N}_{NN} = (\mathcal{U}, A, w)$.

In the case of equidistant pairs of units we have to decide – or to include them all in the graph, or specify an additional selection rule. We shall denote by \mathbf{G}_{NN}^* the graph with included all equidistant pairs, and by \mathbf{G}_{NN} a graph where a single nearest neighbor is always selected.



Structure and properties of the nearest neighbor graphs

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

Let $\mathcal{N}_{NN} = (\mathcal{U}, A, w)$ be a nearest neighbor network. A pair of units $X, Y \in \mathcal{U}$ are *reciprocal nearest neighbors* or RNNs iff $(X, Y) \in A$ and $(Y, X) \in A$.

Suppose $\text{card}(\mathcal{U}) > 1$ and R has no isolated units. Then in \mathcal{N}

- every unit/vertex $X \in \mathcal{U}$ has the $\text{outdeg}(X) \geq 1$ — there is no isolated unit;
- along every walk the values of w are not increasing.

using these two observations we can show that in \mathcal{N}_{NN}^* :

- all the values of w on a closed walk are the same and all its arcs are reciprocal — all arcs between units in a nontrivial (at least 2 units) strong component are reciprocal;
- every maximal (can not be extended) elementary (no arc is repeated) walk ends in a RNNs pair;
- there exists at least one RNNs pair – corresponding to $\min_{X, Y \in \mathcal{U}, X \neq Y} d(X, Y)$.



Fast agglomerative clustering algorithms

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Any network \mathcal{N}_{NN} is a subnetwork of \mathcal{N}_{NN}^* . Its connected components are directed (acyclic) trees with a single RNNs pair in the root.

Based on the nearest neighbor network very efficient algorithms for agglomerative clustering for methods with the reducibility property can be built.

```
chain := [ ]; W := U;  
while card((W)) > 1 do begin  
    if chain = [] then select an arbitrary unit X ∈ W else X := last(chain);  
    grow a NN-chain from X until a pair (Y, Z) of RNNs are obtained;  
    agglomerate Y and Z:  
        T := Y ∪ Z; W := W \ {Y, Z} ∪ {T}; compute D(T, W), W ∈ W  
end;
```

It can be shown that if the clustering method has the reducibility property then the NN-chain remains a NN-chain also after the agglomeration of the RNNs pair.

Network/Create Hierarchy/Clustering with Relational
Constraint/



Example: Slovenian communes

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

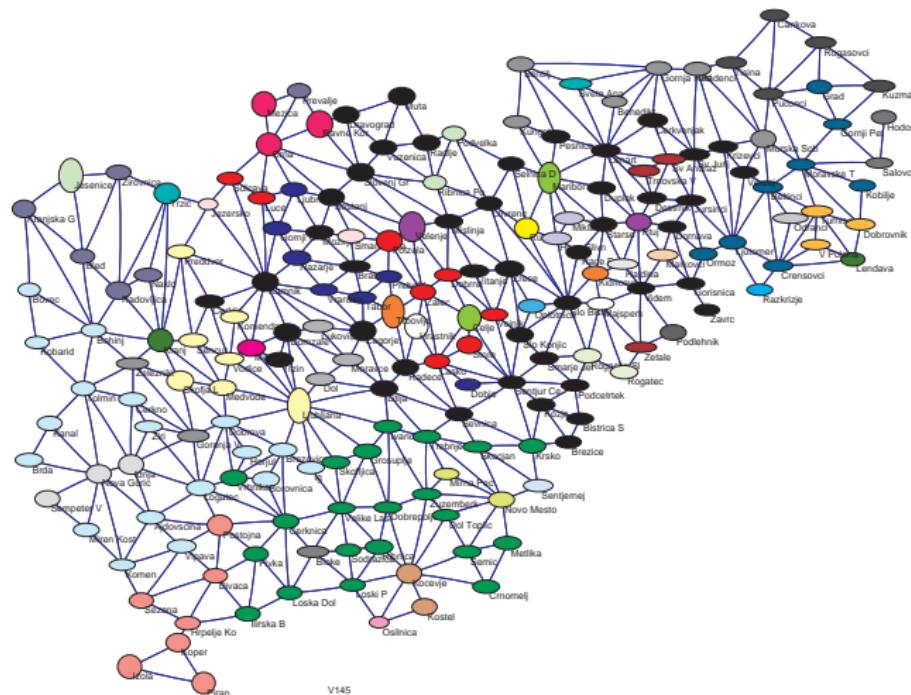
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Example: US counties $t = 1400$

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

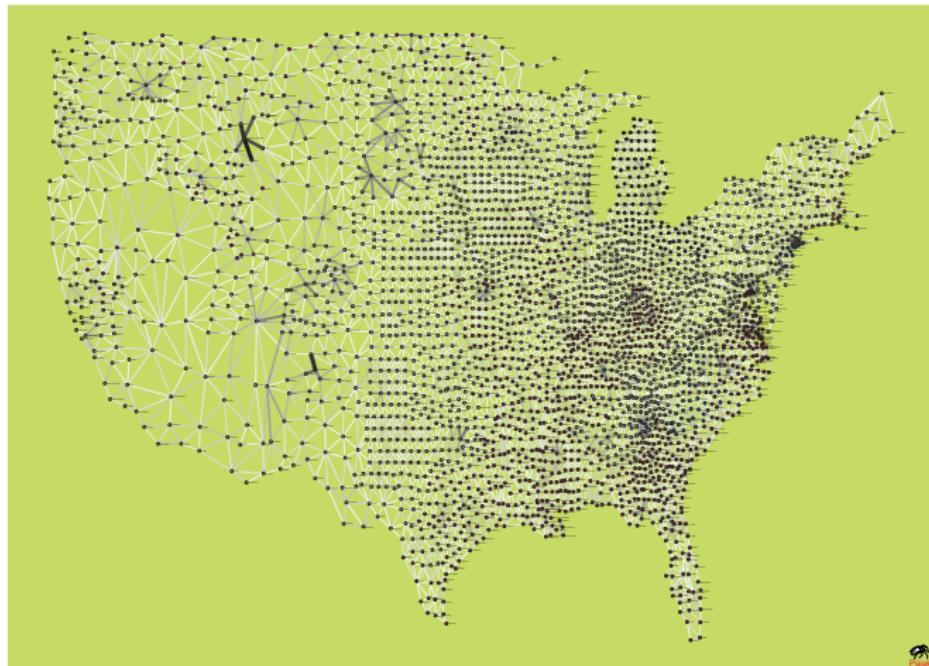
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Example: US counties $t = 200$

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

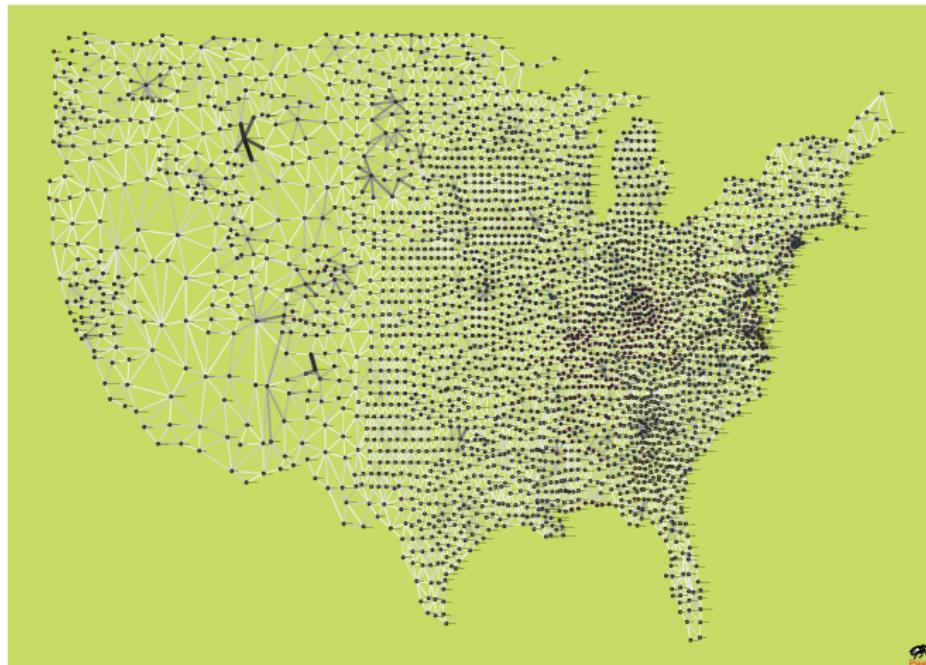
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Louvain method and VOS

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

One of the approaches to determine the clustering \mathbf{C} of a network is to maximize its *modularity*

$$Q(\mathbf{C}) = \sum_{C \in \mathbf{C}} \left(\frac{l(C)}{m} - \left(\frac{d(C)}{2m} \right)^2 \right)$$

where $l(C)$ is the number of edges between nodes belonging to cluster C , and $d(C)$ is the sum of the degrees of nodes from C .

The modularity maximization problem is NP complete.

Louvain method and VOS.

Network/Create Partition/Communities/Louvain Method

Network/Create Partition/Communities/VOS Clustering

[Draw] Layout/VOS Mapping



References

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

- Batagelj V., Ferligoj A. (2000): Clustering relational data. Data Analysis (Eds.: W. Gaul, O. Opitz, M. Schader), Springer, Berlin, 3–15.
- Bruynooghe, M. (1977), Méthodes nouvelles en classification automatique des données taxinomiques nombreuses. Statistique et Analyse des Données, **3**, 24–42.
- Doreian, P., Batagelj, V., Ferligoj, A. (2000), *Symmetric-acyclic decompositions of networks*. *J. classif.*, **17**(1), 3–28.
- Ferligoj A., Batagelj V. (1982), Clustering with relational constraint. *Psychometrika*, **47**(4), 413–426.
- Ferligoj A., Batagelj V. (1983), Some types of clustering with relational constraints. *Psychometrika*, **48**(4), 541–552.
- Murtagh, F. (1985), Multidimensional Clustering Algorithms, *Compstat lectures*, **4**, Vienna: Physica-Verlag.