Network Analysis
Introduction to Networks

Vladimir Batagelj
University of Ljubljana

ECPR Summer School, July 19 – August 4, 2007
Faculty of Social Sciences, University of Ljubljana

Mark Lombardi
Outline

1. Some names in the development of SNA ........................................... 1
2. Some important events ................................................................. 2
3. Selected Books on SNA ................................................................. 3
4. Courses on NA ........................................................................... 4
5. Software for SNA .......................................................................... 5
6. Roman roads (Peutinger) ............................................................... 6
7. Moreno: Who shall survive? ........................................................... 7
8. Development of DNA (Garfield) ..................................................... 9
9. Organic molecule 3CRO ............................................................... 10
10. Hijackers (Krebs) ....................................................................... 11
11. Wall Street Follies ...................................................................... 12
12. They Rule .................................................................................... 13
13. Lombardi’s networks ................................................................. 14
14. Networks .................................................................................... 18
15. Graph ......................................................................................... 19
16. How to get a network? ............................................................... 21
17. Complete and ego-centered networks ........................................ 22
<table>
<thead>
<tr>
<th>Page</th>
<th>Topic</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>Use of existing network data</td>
<td>23</td>
</tr>
<tr>
<td>25</td>
<td>Genealogies</td>
<td>25</td>
</tr>
<tr>
<td>28</td>
<td>Molecular networks</td>
<td>28</td>
</tr>
<tr>
<td>29</td>
<td>GraphML</td>
<td>29</td>
</tr>
<tr>
<td>32</td>
<td>Approaches to computer-assisted text analysis</td>
<td>32</td>
</tr>
<tr>
<td>42</td>
<td>Neighbors</td>
<td>42</td>
</tr>
<tr>
<td>45</td>
<td>Transformations</td>
<td>45</td>
</tr>
<tr>
<td>46</td>
<td>Networks from the Internet</td>
<td>46</td>
</tr>
<tr>
<td>51</td>
<td>Random networks</td>
<td>51</td>
</tr>
</tbody>
</table>
Some names in the development of SNA

Graph theory: Euler, Hamilton, Kirchoff, Kekule, Ford and Fulkerson, Harary, Berge, …

- Moreno (1934) – sociometry
- Lewin (1936)
- Warner and Lunt (1941)
- Heider (1946)
- Bavelas (1948) – centrality
- Homans (1950)
- Cartwright and Harary (1956)
- Nadel (1957) – social structure, social positions, roles
- Mitchell (1969)

Some important events

- International Association of Social Network Analysis – INSNA, 1978
- Journal: Social Networks, 1978
- Newsletter: Connections, 1978
- SUNBELT conferences, 1981
Selected Books on SNA

Courses on NA

- Steve Borgatti, UCINET
- Barry Wellman, University of Toronto
- Douglas White, University of California Irvine
- Lada Adamic, University of Michigan
- James Moody, Duke University
- Mark Newman, University of Michigan
- Jon Kleinberg, Cornell University
- Robert A. Hanneman, University of California, Riverside; workshop
- Noah Friedkin, University of California, Santa Barbara
- John Levi Martin, University of Wisconsin, Madison
- Vladimir Batagelj, University of Ljubljana
- Andrej Mrvar, University of Ljubljana
Software for SNA

UCINET, NetDraw  Pajek  Netminer
Visone  SNA/R  StOCNET
Negopy  InFlow  GUESS
NetworkX  prefuse  JUNG
BGL/Python

See also the INSNA list and recent overview by M. Huisman and M.A.J. van Duijn.

Visual Complexity
Roman roads (Peutinger)
Moreno: Who shall survive?
Sociograph

Figure 3. Group 11A. A social status sociograph of the 68 boys of the 11th grade. It shows quintile positions for: (1) grade average, (2) social participation scores in student organizations, and (3) total friendship choices on a sociometric test (total choices include direct and indirect choices). The student's position on the chart is that of his clique group average for both social participation scores and total friendship choices. Out-of-class choices appear in the side margin; out-of-school choices in the lower margin.
In 1964 E. Garfield with collabora-
tors produced, on the basis of the 
book Asimov I.: *The Genetic Code*
(1963), a corresponding ’citation’ 
network. It was shown that the anal-
ysis ’demonstrated a high degree of 
coincidence between an historian’s 
account of events and the citational 
relationship between these events’.
Organic molecule 3CRO
This dense under-layer of prior trusted relationships made the hijacker network both stealth and resilient. Although we don’t know all of the internal ties of the hijackers’ network, it appears that many of the ties were concentrated around the pilots. This is a risky move for a covert network. Concentrating both unique skills and connectivity in the same nodes makes the network easier to disrupt – once it is discovered. Peter Klerks (Klerks 2001) makes an excellent argument for targeting those nodes in the network that have unique skills. By removing those necessary skills from the project, we can inflict maximum damage to the project mission and goals. It is possible that those with unique skills would also have unique ties within the network. Because of their unique human capital and their high social capital, the pilots were the richest targets for removal from the network. Unfortunately, they were not discovered in time.
Wall Street Follies
They Rule
Lombardi’s networks

Mark Lombardi
(1951-2000)
transformed business relations into art.
FAS: The scientific field of Austria

THE SCIENTIFIC FIELD OF AUSTRIA

Network based on the co-occurrence of assigned scientific classification codes in 5217 projects funded by the Austrian Sciences Fund (FWF) from 1994 to April 2004.
Katy Börner: Text analysis
Ulrik Brandes: Discourse network
A network is based on two sets – set of vertices (nodes), that represent the selected units, and set of lines (links), that represent ties between units. They determine a graph. A line can be directed – an arc, or undirected – an edge.

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

Network = Graph + Data

The data can be measured or computed.
Graph

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

arc = directed line, \((a, d)\)
a is the initial vertex,
d is the terminal vertex.

edge = undirected line, \((c: d)\)
c and d are end vertices.
Networks / Formally

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

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

- *vertex value functions* / properties: $p : \mathcal{V} \rightarrow \mathcal{A}$

- *line value functions* / weights: $w : \mathcal{L} \rightarrow \mathcal{B}$
How to get a network?

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

How to measure networks (questionaires, interviews, observations, archive records, experiments, \ldots)?

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

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.
Complete and ego-centered networks

COMPLETE NETWORK

EGO-CENTERED NETWORKS

Egos  Alters
Use of existing network data

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

Genealogies

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.

Several programs are available for preparation and maintainance of genealogies: free GIM and commercial Brothers Keeper (Slovenian version is available at SRD).

From the data collected in Phd. thesis:
the Ragusa network was produced.
GEDCOM

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

```
0 HEAD
1 FILE ROYALS.GED
...
0 @I58@ 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 @F14@
...
0 @I65@ 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 @I58@
1 WIFE @I65@
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

In usual *Ore* graph every person is represented with a vertex; 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 *p-graph* the vertices 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

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.

virus 1GDY: \( n = 39865, m = 40358 \)
GraphML

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.net
java org.apache.xalan.xslt.Process -IN p.xml -XSL m.xsl -OUT p.txt

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="v1" target="v4"/>
    <edge source="v1" target="v6"/>
    <edge source="v1" target="v10"/>
    <edge source="v1" target="v12"/>
    <edge source="v2" target="v1"/>
    <edge source="v2" target="v5"/>
    <edge source="v2" target="v8"/>
    <edge source="v2" target="v11"/>
    <edge source="v3" target="v1"/>
    <edge source="v3" target="v4"/>
    <edge source="v3" target="v7"/>
    <edge source="v3" target="v9"/>
    <edge source="v3" target="v12"/>
    <edge source="v4" target="v1"/>
    <edge source="v4" target="v5"/>
    <edge source="v4" target="v11"/>
    <edge source="v4" target="v12"/>
    <edge source="v5" target="v1"/>
    <edge source="v5" target="v2"/>
    <edge source="v5" target="v6"/>
    <edge source="v5" target="v12"/>
    <edge source="v6" target="v1"/>
    <edge source="v6" target="v2"/>
    <edge source="v6" target="v4"/>
    <edge source="v6" target="v11"/>
    <edge source="v6" target="v12"/>
    <edge source="v7" target="v3"/>
    <edge source="v7" target="v11"/>
    <edge source="v8" target="v3"/>
    <edge source="v8" target="v5"/>
    <edge source="v8" target="v11"/>
    <edge source="v8" target="v12"/>
    <edge directed="false" source="v2" target="v5"/>
    <edge directed="false" source="v2" target="v6"/>
    <edge directed="false" source="v3" target="v4"/>
    <edge directed="false" source="v3" target="v7"/>
    <edge directed="false" source="v5" target="v7"/>
    <edge directed="false" source="v6" target="v8"/>
    <edge directed="false" source="v6" target="v9"/>
    <edge directed="false" source="v11" target="v12"/>
    <edge directed="false" source="v12" target="v8"/>
  </graph>
</graphml>
GraphML → Pajek using XSLT

```xml
<?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> * Edges </xsl:text>
    <xsl:apply-templates select="graphml/graph/edge" />
    <xsl:text> * Arcs </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>"&gt;</xsl:text>
  </xsl:template>
</xsl:stylesheet>
```
Approaches to computer-assisted text analysis


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 mainly 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.
…approaches to CaTA

In thematic TA the units are coded as rectangular matrix

Text units × 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.

This coding can be directly considered as network with Subjects ∪ Objects
as vertices and lines labeled with Verbs.

See also RDF triples in semantic web.
Network CaTA

This way we already steped into the network TA.
Examples:
Carley: Cognitive maps,
J.A. de Ridder: CETA,
Megaputer: Text Analyst.


There are additional ways to obtain networks from textual data.
TA – Dictionary networks

In a dictionary graph the terms determine the set of vertices, 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).


Artlex, Wordnet, ConceptNet, OpenCyc.

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

Paper.
TA – Citation networks

In a *citation graph* the vertices are different publications from the selected area; two publications are connected by an arc if the first is cited by the second. The citation networks are almost acyclic.

E. Garfield: HistCite / Pajek, papers.
An example of very large citation network is US Patents / Nber, 
\( n = 3774768, m = 16522438 \).
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.
TA – International Relations

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.
Recoding of KEDS/WEIS data in Pajek’s format

% 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"
... 223 "MIL ENGAGEMENT"
*arcs :224 "RIOT"
*arcs :225 "ASSASSINATE TORTURE"
212: 314 83 1 [4] 890404 YUG ETHALB 212 (ARREST PERSON) ALB ETHNIC JAILED IN YUG
224: 3 83 1 [4] 890407 ALB ETHALB 224 (RIOT) RIOTS
105: 105 63 1 [175] 030731 GER CYP 042 (ENDORSE) GAVE SUPPORT
212: 295 35 1 [175] 030731 UNWCT BOSSER 212 (ARREST PERSON) SENTENCED TO PRISON
43: 306 87 1 [175] 030731 VAT EUR 043 (RETRACT) RALLIED
13: 295 35 1 [175] 030731 UNWCT BOSSER 013 (RETRACT) CLEARED
121: 295 22 1 [175] 030731 UNWCT BAL 121 (CRITICIZE) CHARGES
122: 246 295 1 [175] 030731 SER UNWCT 122 (DENIGRATE) TESTIFIED
121: 295 22 1 [175] 030731 BOSSER UNWCT 121 (CRITICIZE) ACCUSED
...Recoding programs in R

To recode the KEDS/WEIS data we used short programs in R, such as the following one:

```r
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 + 100000
      if (t<0) t0 <<- t; u <- z[2]; v <- z[3]; r <- z[4]
      if (is.na(as.numeric(r))) cat(nlin,'NA rel-code',r,'

      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)
    }
  }
}
```

---

ECPR Summer School, Ljubljana, July 19 – August 4, 2007
Recoding programs in R

```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("% Recoded by WEISmonths,"date()
",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], '" [1- * \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: The dictionary data structure is in R implemented as `environment`.
Neighbors

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: \( \mathcal{N}(k) = (\mathcal{V}, A, d) \)

\[(u, v) \in A \iff v \text{ is among } k \text{ nearest neighbors of } u, \quad w(u, v) = d(u, v)\]

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

\[(u : v) \in \mathcal{E} \iff d(u, v) \leq r, \quad w(u, v) = w(v, u) = d(u, v)\]

These networks provide a link between data analysis and network analysis. Efficient algorithms ?!

Fisher’s Iris data.

Details on Multivariate networks and procedures in R.
Nearest $k$ neighbors in R

```r
kneighbor2Net <-
# 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)
}
stand <-
# standardizes vector x .
function(x){
  s <- sd(x)
  if (s > 0) (x - mean(x))/s else x - x
}
data(iris)
ir <- cbind(stand(iris[,1]),stand(iris[,2]),stand(iris[,3]),
  stand(iris[,4]))
kneighbor2Net("iris5.net",as.matrix(dist(ir)),5)
```
Fisher’s Irises

Draw/Draw-Partition-2Vectors

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

Words graph – words from a given set are vertices; 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 – vertices 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 – vertices are states in the game; two states are linked with an arc if the rules of the game allow the transiton from first to the second state.
Networks from the Internet

Internet Mapping Project. Links among WWW pages.
KartOO, TouchGraph.
Derived from archives of E-mail, blogs, . . . , server’s logs.
Cybergeography, CAIDA.
Collecting Networks from WWW

*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.
Networks from Amazon in R

```r
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('<[/]>','line'); r <- j[1]-1; substr(line,l,r)
}
vid <- new.env(hash=TRUE,parent=emptyenv())
vtx <- file(fvtx,"w"); cat(' * vertices
', file=vtx)
tit <- file(ftit,"w"); cat(' * vertices
', file=tit)
lnk <- file(flnk,"w"); cat(' * arcs
',file=lnk)
url1 <- 'http://www.amazon.com/exec/obidos/tg/detail/-/
url2 <- '?v=glance';
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,'
books <- c(book)
```
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"\n",sep='',file=vtx)
                cat('new vertex ',nver," ",bk," URL ",url1,bk,url2,\n"\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');
        }
    }
    flush.console()
}
close(lnk); close(vtx); cat('Amazon - END\n')
Networks from Amazon – books on SNA

Books in SNA from Amazon, 10. November 2006; Starting point P. Doreian &: Generalized Block-modeling.

SVG picture. Files/ZIP.

The program in R is just a skeleton. Possible improvements: list of starting points; continuation after interrupts; . . .
Random networks

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

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

Available is also a program GeneoRnd for generating random genealogies.
Random undirected graph of Erdős-Rényi type

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

ErdosRenyiNet <-
# generates a random undirected graph of Erdos-Renyi type
# with n vertices 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)
}