Introduction to Network Analysis using **Pajek**

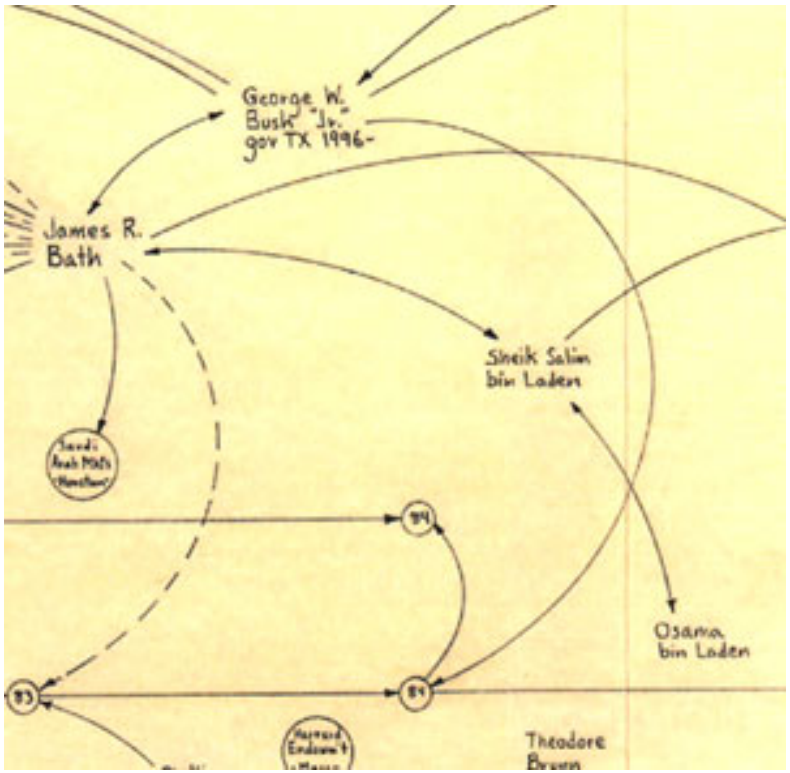
2. Sources of networks

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

PhD and MS program in Statistics
University of Ljubljana, 2022

Outline



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?

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 completness*, and which properties of nodes/links we shall consider.

How to measure networks (questionaires, 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*.

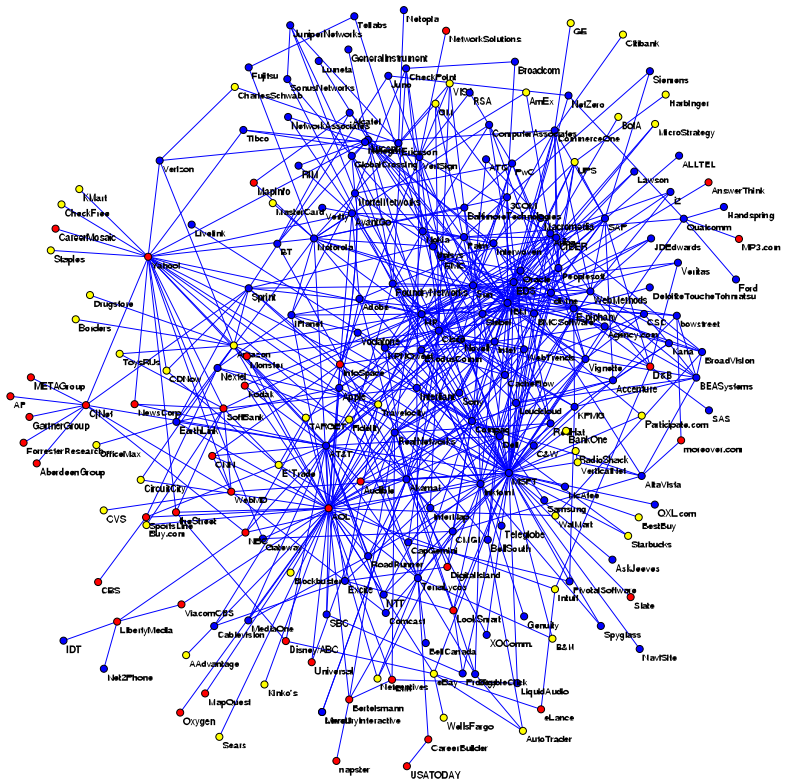
Use of existing network data

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



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

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

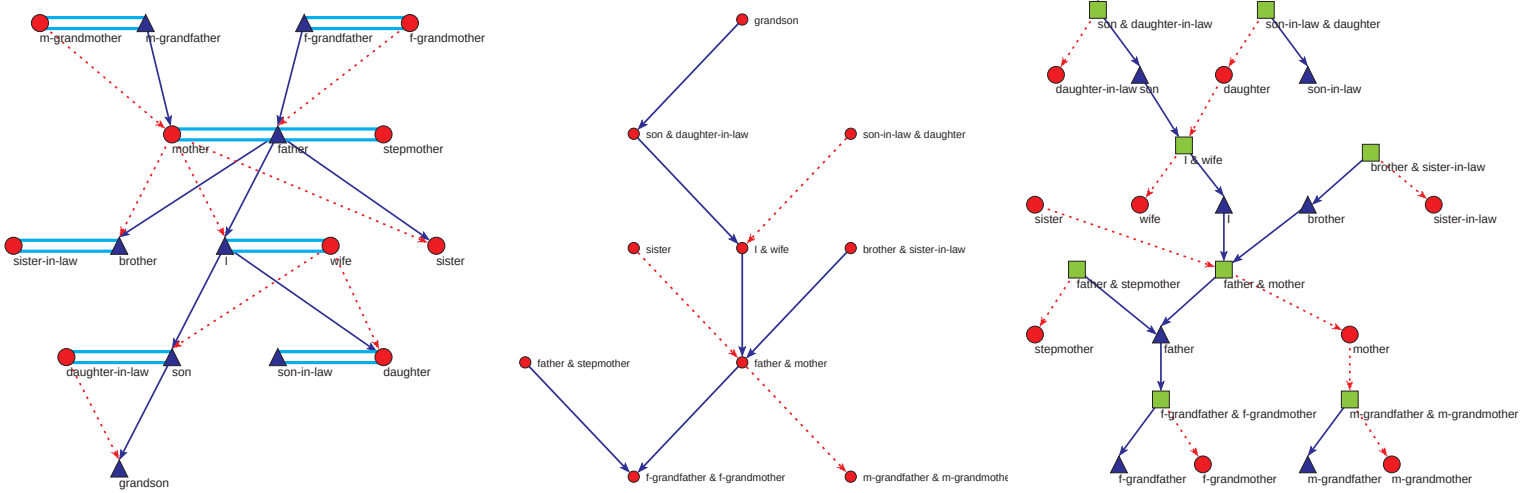
GEDCOM is a standard for storing genealogical data, which is used to interchange and combine data from different programs, which were used for entering the data.

```
0 HEAD
1 FILE ROYALS.GED
...
0 @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
```

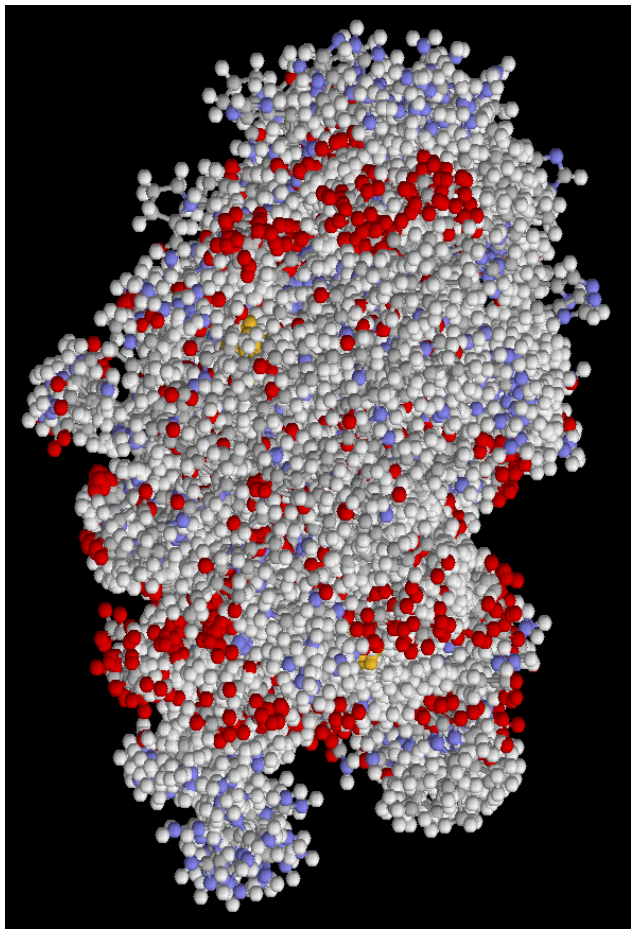
Network representations of genealogies

In 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 dauther of* (dotted red). More about p-graphs *D. White*.



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

Molecular networks



virus 1GDY: $n = 39865$, $m = 40358$

In the Brookhaven Protein Data Bank we can find many large organic molecula (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

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>
```

```
*Vertic
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

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 semiautomaticly. As *units* of TA we usually consider clauses, statements, paragraphs, news, messages, ...

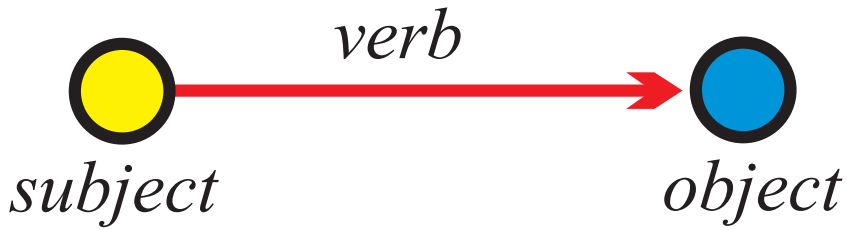
Solutions for names: ResearcherID, ORCID, AMS; for words: dictionaries, stemming, lemmatization.

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

. . . approaches to CaTA

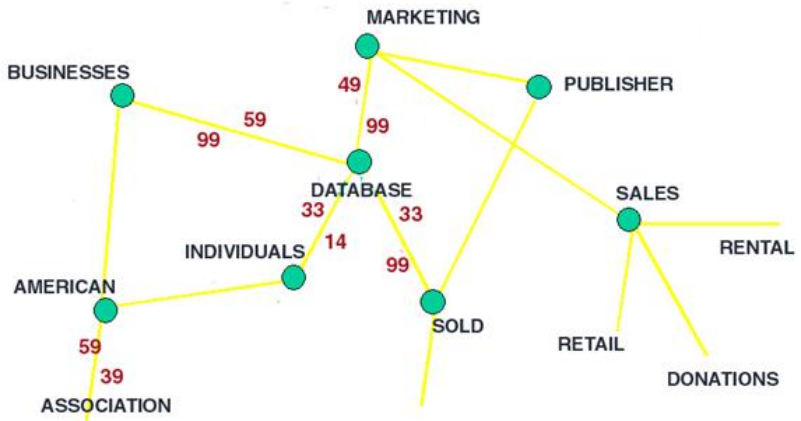
In thematic TA the units are coded as rectangular matrix *Text units* $\times$ *Concepts* which can be considered as a two-mode network.
Examples: M.M. Miller: VBPro, H. Klein: Text Analysis/ TextQuest.

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



TextAnalyst's 'semantic network'

This way we already steped 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

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


```
% 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*

. . . Program in R


```
# 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],'" [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]),' "',
  get(b[i],env=rel,inherits=FALSE),'"\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

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 nonperiodical 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

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.

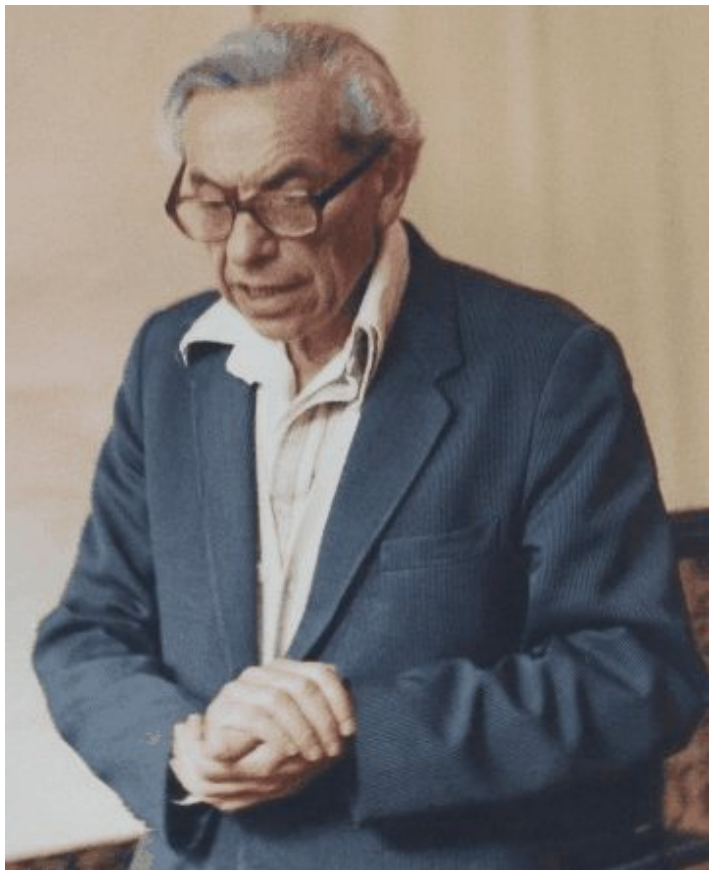
Online Dictionary of Library and Information Science *ODLIS*, *Odlis.net* (2909 / 18419).

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

Artlex, *Wordnet*, *ConceptNet*, *OpenCyc*.

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

Collaboration networks



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

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, $\quad w(u,v) = d(u,v$

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

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

These networks provide a link between data analysis and network analysis. Efficient algorithms ?! Nearest neighbor library in R-package yaImpute.
Fisher's *Iris data*. Details on Multivariate networks and procedures in R.

Nearest *k* neighbors in R

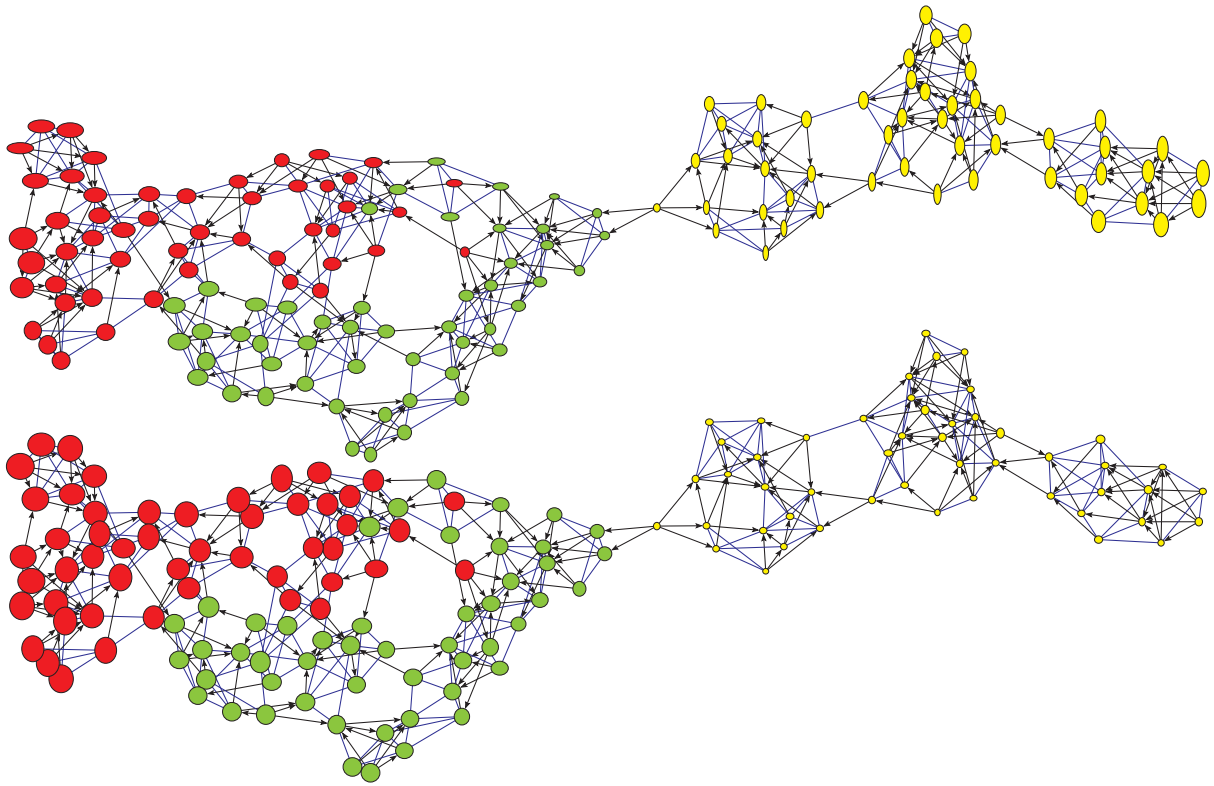

```
k.neighbor2Net <-
# stores network of first  k  neighbors for
# dissimilarity matrix  d  to file  fnet  in Pajek format.
function(fnet,d,k){
  net <- file(fnet,"w")
  n <- nrow(d); rn <- rownames(d)
  cat("*vertices",n,"\n",file=net)
  for (i in 1:n) cat(i," \"",rn[i],"\"\n",sep="",file=net)
  cat("*arcs\n",file=net)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

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



`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


```
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

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 transiton 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

KartOO network

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*.

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

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.
Nutch, IssueCrawler, W4F.
Python: lxml; Beautiful Soup.
Amazon web services, Google Data, Google+, YouTube, Twitter, Last.fm, MusicBrainz3, Flickr, LinkedIn, . . .

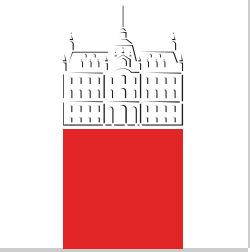
Networks from Amazon in R

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

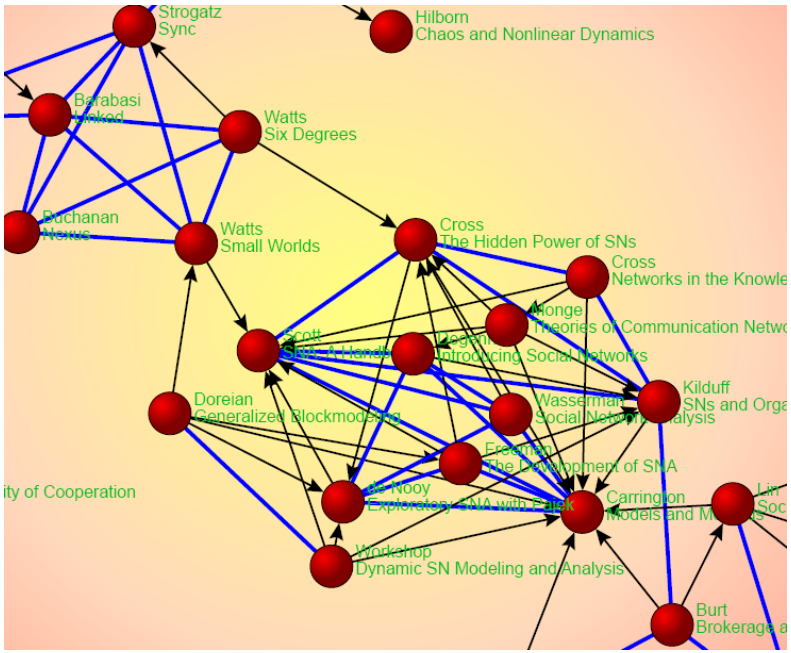
```
amazon <- function(fvtx,flnk,ftit,maxver){
# Creates a network of books from Amazon
#  amazon('v.txt','a.txt','t.txt',10)
# Vladimir Batagelj, 20-21. nov. 2004 / 10. nov. 2006
  opis <- function(line){
    i <- regexpr('\">',line); l <- i[1]+attr(i,"match.length")[1
    j <- regexpr('</a>',line); 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


```
  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



Books in SNA from Amazon, 10. november 2006; Starting point P. Doreian &: Generalized Blockmodeling.

The program in R is just a skeleton. Possible improvements: list of starting points; continuation after interrupts; etc.
The structure of Amazon files is changing!!!

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 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


```
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)
}
```