Photo: V. Batagelj, ***Araneus diadematus***

Network analysis of KEDS

Vladimir Batagelj
Andrej Mrvar

University of Ljubljana

Viszards session

Sunbelt XXV, Redondo Beach, CA, February 19, 2005

version: February 19, 2005 / 20 : 06

Outline

Transforming KEDS into Pajek's format

KEDS (WEIS) data determine a *temporal network*. In it the presence/activity of vertex/link can change through time. **Pajek** supports two types of descriptions of temporal networks based on *presence* and on *events*. For our purposes we used the presence based notation:

```
*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]
```

Vertex a is present in time intervals 5 to 10 and 12 to 14.

Edge $(1 : 3)$ is present in time intervals 6 to 8.

A link is present, if both its end-vertices are present.

To enable encoding of KEDS (WEIS) data into Pajek's format we had to implement in Pajek also a support of multiple (multi-relational) networks. This was done in November 2004.

Multiple networks

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 endpoints 47 and 14 and weight 5 belongs to relation 3.

KEDS in Pajek's format.

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"
 ...
*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 ETHNIC JAILED IN YUG
224: 3 83 1 [4]                 890407  ALB     ETHALB  224  (RIOT)  RIOTS
123: 83 153 1 [4]               890408  ETHALB  KSV     123  (INVESTIGATE)   PROBING
 ...
42: 105 63 1 [175]              030731  GER     CYP     042  (ENDORSE)       GAVE SUPPORT
212: 295 35 1 [175]             030731  UNWCT   BOSSER  212  (ARREST PERSON) SENTENCED TO PRISON
43: 306 87 1 [175]              030731  VAT     EUR     043  (RALLY) RALLIED
13: 295 35 1 [175]              030731  UNWCT   BOSSER  013  (RETRACT)       CLEARED
121: 295 22 1 [175]             030731  UNWCT   BAL     121  (CRITICIZE)     CHARGES
122: 246 295 1 [175]            030731  SER     UNWCT   122  (DENIGRATE)     TESTIFIED
121: 35 295 1 [175]             030731  BOSSER  UNWCT   121  (CRITICIZE)     ACCUSED
```

. . . Recoding programs in R

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

```
# 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)
    }
  }
```

. . . Recoding programs 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: The dictionary data structure is in R implemented as *environment*.

. . . Recoding programs in R

We produced different recodings: by months, by day and by the day of the week. The following list represents numbering of months for Balkan data.

		21	900901	41	920501	61	940101	81	950901	101	970501	121	990102	141	000901	161	020503
		22	901004	42	920601	62	940201	82	951001	102	970601	122	990201	142	001001	162	020601
		23	901102	43	920701	63	940301	83	951101	103	970701	123	990301	143	001101	163	020701
4	890402	24	901203	44	920801	64	940401	84	951201	104	970801	124	990401	144	001201	164	020801
5	890504	25	910104	45	920901	65	940501	85	960101	105	970901	125	990501	145	010101	165	020901
6	890601	26	910202	46	921002	66	940601	86	960201	106	971001	126	990601	146	010201	166	021001
7	890706	27	910301	47	921101	67	940701	87	960301	107	971101	127	990701	147	010301	167	021101
8	890801	28	910401	48	921201	68	940801	88	960401	108	971202	128	990801	148	010401	168	021201
9	890901	29	910502	49	930101	69	940901	89	960501	109	980105	129	990901	149	010501	169	030101
10	891004	30	910602	50	930201	70	941001	90	960601	110	980201	130	991001	150	010601	170	030201
11	891102	31	910701	51	930301	71	941101	91	960701	111	980302	131	991101	151	010701	171	030303
		32	910801	52	930401	72	941201	92	960801	112	980401	132	991201	152	010801	172	030401
13	900101	33	910901	53	930501	73	950101	93	960901	113	980501	133	000101	153	010901	173	030501
14	900201	34	911001	54	930601	74	950201	94	961001	114	980601	134	000201	154	011001	174	030601
15	900301	35	911101	55	930701	75	950301	95	961101	115	980701	135	000301	155	011101	175	030701
16	900402	36	911201	56	930801	76	950401	96	961201	116	980801	136	000401	156	011201		
17	900501	37	920101	57	930901	77	950501	97	970101	117	980901	137	000501	157	020101		
18	900611	38	920201	58	931001	78	950601	98	970201	118	981001	138	000601	158	020201		
19	900701	39	920301	59	931101	79	950701	99	970301	119	981101	139	000701	159	020301		
20	900802	40	920401	60	931201	80	950801	100	970401	120	981201	140	000801	160	020401		

Examples on the computer

Standard approaches:

- layout of the entire network using spring embedder
- sequence of time slices
- selected relation

We get a 'rainbow'. Difficult to see something.

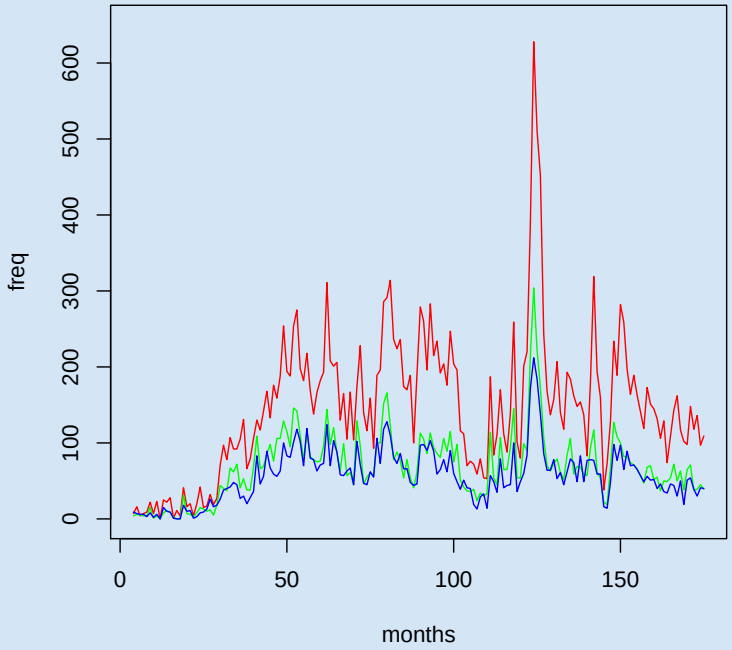
Positive, neutral, negative relations

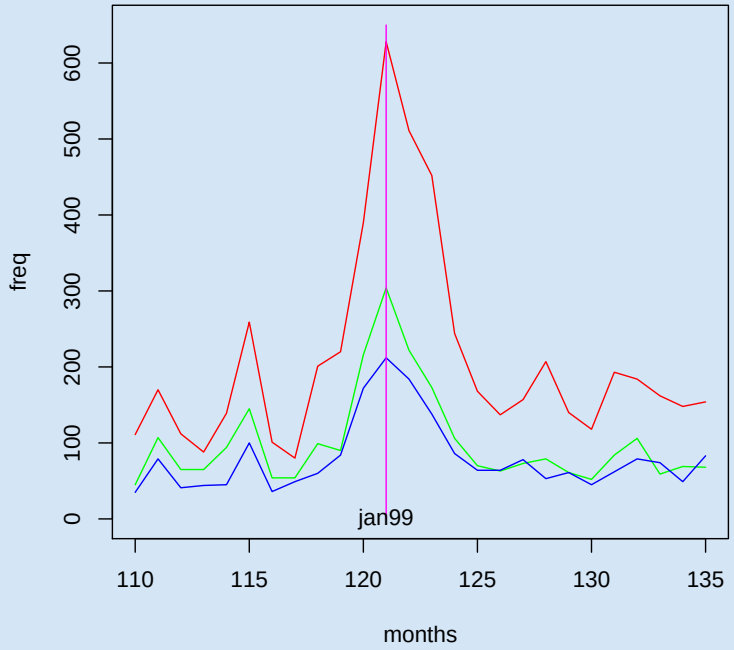
We decided to merge actions into 3 groups

Positive	Neutral	Negative
01 Yield	08 Agree	15 Demand
02 Comment	09 Request	16 Warn
03 Consult	10 Propose	17 Threaten
04 Approve	11 Reject	18 Demonstrate
05 Promise	12 Accuse	19 Reduce Relationship
06 Grant	13 Protest	20 Expel
07 Reward	14 Deny	21 Seize
		22 Force

Some results

Time changing of numbers of links. Repetitive operations !!!



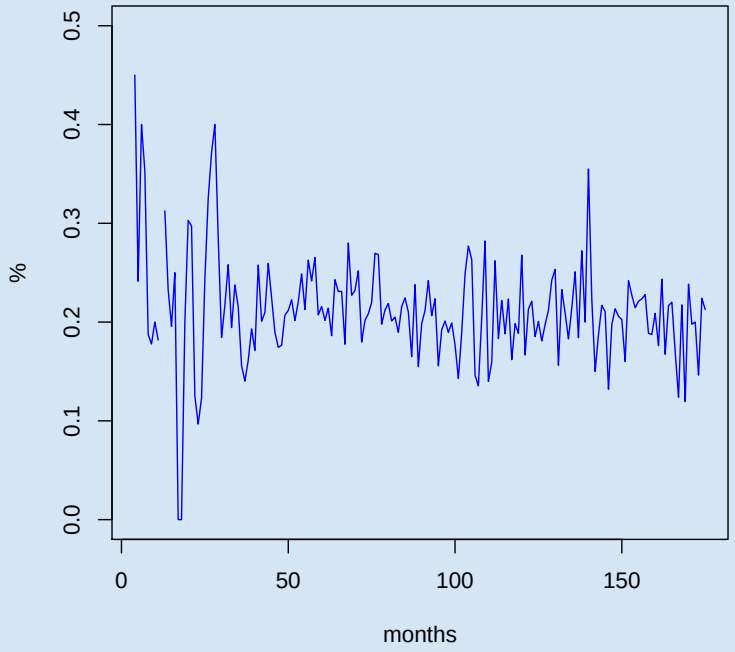



```
months <- 4:175
plot(months,v3,type="l",ylim=c(0,650),ylab="freq",xlab="months",col="red")
lines(months,v2,col="green"); lines(months,v1,col="blue")

m <- 110:135
plot(m,v3[m],type="l",ylim=c(0,650),ylab="freq",xlab="months",col="red")
lines(m,v2[m],col="green"); lines(m,v1[m],col="blue")
t <- 121; lines(c(t,t),c(0,650),col="magenta"); text(t,0,"jan99")
```

. . . Some results

Relative frequencies of negative ties




```
v <- v1/(v1+v2+v3)
plot(months,v,ylim=c(0,0.5),ylab="%",type="l",col="blue")
```

. . . Some results

Pictures through time. Reduced by less frequent.

Neighborhoods of selected vertices.

Pre, at, post events pictures.

Plans for the future analyses

An interesting approach could be search for typical temporal patterns – stories in the network. In Pajek a pattern search is implemented for ordinary networks. For this purpose we intend to extend it also to temporal patterns.