Link Analysis with Pajek

Vladimir Batagelj
Nataša Kejžar
Andraž Mrvar
Matjaž Zaveršnik
University of Ljubljana

Acai-05 Advanced Course on Knowledge Discovery
Ljubljana, Slovenia, June 27-July 5, 2005
Outline

1 Networks ................................................................. 1
5 Multiple networks ...................................................... 5
7 Temporal networks ...................................................... 7
12 Statistics ................................................................. 12
17 Decompositions ......................................................... 17
18 Cuts .................................................................. 18
20 Citation weights ......................................................... 20
21 Cores and generalized cores ........................................ 21
22 Islands ................................................................ 22
28 Triangular connectivity and triangular networks ............... 28
32 Pattern searching ......................................................... 32
39 Temporal Analysis of US Patents Network ....................... 39
50 PajektoSVGanim ......................................................... 50
51 Further Readings / Books .............................................. 51
Further Readings / Papers ......................................................... 53
Some links .................................................................................. 55
Networks

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 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 links. $n = \text{card}(\mathcal{V})$, $m = \text{card}(\mathcal{L})$

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

- $\mathcal{W}$ line value functions / weights: $w : \mathcal{L} \rightarrow B$

In November 1996 we started the development of Pajek – a program, for analysis and visualization of large networks. The latest version of Pajek is freely available, for noncommercial use, at its home page:

http://vlado.fmf.uni-lj.si/pub/networks/pajek/
Large Networks

Large network – several thousands or millions of vertices.

Usually sparse $m \ll n^2$; typical: $m = O(n)$ or $m = O(n \log n)$.

Examples:

| network                | size | $n = |V|$ | $m = |L|$ | source                            |
|------------------------|------|---------|---------|-----------------------------------|
| ODLIS dictionary       | 61K  | 2909    | 18419   | ODLIS online                      |
| Citations SOM          | 168K | 4470    | 12731   | Garfield’s collection             |
| Molecula 1ATN          | 74K  | 5020    | 5128    | Brookhaven PDB                    |
| Comput. geometry       | 140K | 7343    | 11898   | BiBTeX bibliographies             |
| English words 2-8      | 520K | 52652   | 89038   | Knuth’s English words             |
| Internet traceroutes   | 1.7M | 124651  | 207214  | Internet Mapping Project          |
| Franklin genealogy     | 12M  | 203909  | 195650  | Roperld.com gedcoms               |
| World-Wide-Web         | 3.6M | 325729  | 1497135 | Notre Dame Networks               |
| Actors                 | 3.9M | 392400  | 1342595 | Notre Dame Networks               |
| US patents             | 82M  | 3774768 | 16522438| Nber                              |
| SI internet            | 38M  | 5547916 | 62259968| Najdi Si                          |
Pajek’s data types

In Pajek analysis and visualization are performed using 6 data types: network, partition, vector, cluster, permutation, and hierarchy.

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.
**Example: Wolfe Monkey Data**

<table>
<thead>
<tr>
<th>inter.net</th>
<th>inter.net</th>
<th>sex.clu</th>
<th>age.vec</th>
<th>rank.per</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Vertices 20</td>
<td>1   6   5</td>
<td>*vertices 20</td>
<td>15   10   8</td>
<td></td>
</tr>
<tr>
<td>1 &quot;m01&quot;</td>
<td>1   7   9</td>
<td>1   10   1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 &quot;m02&quot;</td>
<td>1   8   7</td>
<td>1   10   1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 &quot;m03&quot;</td>
<td>1   9   4</td>
<td>1   8   4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 &quot;m04&quot;</td>
<td>1  10   3</td>
<td>1   7   5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 &quot;m05&quot;</td>
<td>1   11   2</td>
<td>15  10   11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 &quot;f06&quot;</td>
<td>1   12   7</td>
<td>11   5   6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 &quot;f07&quot;</td>
<td>1   13   3</td>
<td>11   5   6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 &quot;f08&quot;</td>
<td>1   14   2</td>
<td>12   9   9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 &quot;f09&quot;</td>
<td>1   15   5</td>
<td>12   9   9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 &quot;f10&quot;</td>
<td>1   16   1</td>
<td>16   7   8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 &quot;f11&quot;</td>
<td>1   17   4</td>
<td>10  14  18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 &quot;f12&quot;</td>
<td>1   18   1</td>
<td>14   5  19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 &quot;f13&quot;</td>
<td>2   3   5</td>
<td>7   20  20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14 &quot;f14&quot;</td>
<td>2   4   1</td>
<td>11  13  13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 &quot;f15&quot;</td>
<td>2   5   3</td>
<td>7   14  14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 &quot;f16&quot;</td>
<td>2   6   1</td>
<td>5   15  15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17 &quot;f17&quot;</td>
<td>2   7   4</td>
<td>15  16  16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 &quot;f18&quot;</td>
<td>2   8   2</td>
<td>4   17  17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19 &quot;f19&quot;</td>
<td>2   9   6</td>
<td>4   17  17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 &quot;f20&quot;</td>
<td>2  10   2</td>
<td>4   17  17</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Edges*

| 1   2   2 | 2   11   5 |
| 1   3  10 | 2  12   4 |
| 1   4   4 | 2  13   3 |
| -   -   - | 2   -   - |

...
Multiple networks

*Multiple* or *multi-relational* networks on the same set of vertices were implemented in *Pajek* only recently (November 2004). 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, . . . ), . . .
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 / Gulf
Temporal networks

In a *temporal network* 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*.
Temporal networks – presence

*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.
## Temporal networks – events

| Event | Explanation | *Vertices 3 *
|-------|-------------|----------------|
| **TI** $t$ | initial events – following events happen when time point $t$ starts | **Events**
| **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 vertex $v$ |
| **SV** $v$ | show vertex $v$ |
| **DV** $v$ | delete vertex $v$ |
| **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 vertex $v$ 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 line $(u,v)$ |
| **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 lines: $k$ denotes the $k$-th line – **HE**: 3 14 37 hides the third edge connecting vertices 14 and 37.

**Time.tim. Friends.tim.**
Temporal networks / September 11

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.

Pictures in SVG: 66 days.
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
  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
  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
Statistics

Input data

- numeric $\rightarrow$ vector
- ordinal $\rightarrow$ permutation
- nominal $\rightarrow$ clustering (partition)

Computed properties

**global**: number of vertices, 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

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 vertices and 325624 arcs (564 loops); number of lines with value=1 is 227481.
... Statistics

To identify the vertices with the largest degree:

Net/Partitions/Degree/All
Partition/Make vector
Info/Vector +10

The largest degrees have the vertices:

<table>
<thead>
<tr>
<th>vertex</th>
<th>deg</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12720</td>
<td>1108</td>
</tr>
<tr>
<td>2</td>
<td>12459</td>
<td>1074</td>
</tr>
<tr>
<td>3</td>
<td>8878</td>
<td>878</td>
</tr>
<tr>
<td>4</td>
<td>18122</td>
<td>875</td>
</tr>
<tr>
<td>5</td>
<td>13793</td>
<td>803</td>
</tr>
<tr>
<td>6</td>
<td>13181</td>
<td>799</td>
</tr>
<tr>
<td>7</td>
<td>23136</td>
<td>732</td>
</tr>
<tr>
<td>8</td>
<td>15080</td>
<td>723</td>
</tr>
<tr>
<td>9</td>
<td>13948</td>
<td>720</td>
</tr>
<tr>
<td>10</td>
<td>22973</td>
<td>716</td>
</tr>
</tbody>
</table>
Statistics / Pajek and R

**Pajek** 0.89 (and higher) supports interaction with statistical program R and the use of other programs as tools.

**Partition/Make Vector**
**Tools/Program R/Send to R/Current Vector**

We send the vector of in-degrees to R and draw their distribution

```r
summary(v2)
t <- tabulate(v2)
c <- t[t>0]
i <- (1:length(t))[t>0]
plot(i,c,log='xy',main='EAT all-degree distribution',
     xlab='deg',ylab='freq')
```

The obtained picture can be saved with **File/Save as** in selected format (PDF or PS for \LaTeX{}; Windows metafile format for inclusion in Word).
EAT all-degree distribution

EAT all-degree distribution

deg
freq
1 5 10 50 100 500
1 5 10 50 500 5000
Decompositions

The main goals in the design of Pajek are:

- to support abstraction by (recursive) decomposition of a large network into several smaller networks that can be treated further using more sophisticated methods;
- to provide the user with some powerful visualization tools;
- to implement a selection of efficient subquadratic algorithms for analysis of large networks.

With Pajek we can: find clusters (components, neighbourhoods of ‘important’ vertices, cores, etc.) in a network, extract vertices that belong to the same clusters and show them separately, possibly with the parts of the context (detailed local view), shrink vertices in clusters and show relations among clusters (global view).
Cuts

The standard approach to find interesting groups inside a network was based on properties/weights – they can be *measured* or *computed* from network structure (for example Kleinberg’s *hubs and authorities*).

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

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

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

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

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

and $\mathcal{V}(\mathcal{L}')$ is the set of all endpoints of the lines from $\mathcal{L}'$. 
Simple analysis using cuts

We look at the components of $N(t)$.

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 vertex/arc-values and the distribution of component sizes and considering additional knowledge on the nature of network or goals of analysis.

We developed some new and efficiently computable properties/weights.
Citation weights

The citation network analysis started in 1964 with the paper of Garfield et al. In 1989 Hummon and Doreian proposed three indices – weights of arcs that are proportional to the number of different source-sink paths passing through the arc. We developed algorithms to efficiently compute these indices. Main subnetwork (arc cut at level 0.007) of the SOM (self-organizing maps) citation network (4470 vertices, 12731 arcs). See paper.
The notion of core was introduced by Seidman in 1983. Vertices belonging to a $k$-core have to be linked to at least $k$ other vertices of the core. A very efficient algorithm exists for determining cores.

The notion of core can be extended to other vertex functions and for several of them the corresponding cores can be efficiently determined.

Figure presents the $p_S$-core at level 46 of the collaboration network (7343 vertices, 11898 edges, edge weight counts the number of common works) in the field of computational geometry.

See paper.
Islands

If we represent a given or computed value of vertices / lines as a height of vertices / lines and we immerse the network into a water up to selected level we get islands. Varying the level we get different islands. Islands are very general and efficient approach to determine the ’important’ subnetworks in a given network.

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

See details.
Using CRA S. Corman and K. Dooley produced the **Reuters terror news network** that is based on all stories released during 66 consecutive days by the news agency Reuters concerning the September 11 attack on the US. The vertices of a network are words (terms); there is an edge between two words iff they appear in the same text unit. The weight of an edge is its frequency. It has $n = 13332$ vertices and $m = 243447$ edges.
Islands – US patents

As an example, let us look at Nber network of US Patents. It has 3774768 vertices and 16522438 arcs (1 loop). We computed SPC weights in it and determined all (2,90)-islands. The reduced network has 470137 vertices, 307472 arcs and for different $k$: $C_2 = 187610$, $C_5 = 8859$, $C_{30} = 101$, $C_{50} = 30$ islands. Rolex

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>139793</th>
<th>29670</th>
<th>9288</th>
<th>3966</th>
<th>1827</th>
<th>997</th>
<th>578</th>
<th>362</th>
<th>250</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>190</td>
<td>125</td>
<td>104</td>
<td>71</td>
<td>47</td>
<td>37</td>
<td>36</td>
<td>33</td>
<td>21</td>
<td>23</td>
</tr>
<tr>
<td>[11]</td>
<td>17</td>
<td>16</td>
<td>8</td>
<td>7</td>
<td>13</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>[21]</td>
<td>12</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>[31]</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[41]</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>[51]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[61]</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[71]</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>
Island size distribution

![Island size distribution graph]

- On the x-axis, size ranges from 1 to 100.
- On the y-axis, frequency ranges from 1 to 10,000.

Advanced Course on Knowledge Discovery, Ljubljana, Slovenia, June 27-July 5, 2005
Main path and main island of Patents
### Table 1: Patents on the liquid-crystal display

<table>
<thead>
<tr>
<th>Patent</th>
<th>Date</th>
<th>Author(s) and Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2544659</td>
<td>Mar 13, 1951</td>
<td>Dreyer. Dichroic compounds and liquid crystal materials containing them</td>
</tr>
<tr>
<td>4082428</td>
<td>Apr 4, 1978</td>
<td>Hsu. Liquid crystal composition and method</td>
</tr>
</tbody>
</table>

### Table 2: Patents on the liquid-crystal display

<table>
<thead>
<tr>
<th>Patent</th>
<th>Date</th>
<th>Author(s) and Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>4083797</td>
<td>Apr 11, 1978</td>
<td>Oh. Nematic liquid crystal compositions and method for preparing same and liquid crystal compositions using same</td>
</tr>
<tr>
<td>4510069</td>
<td>Apr 9, 1985</td>
<td>Eidenschink, et al. Cyclohexane derivatives</td>
</tr>
</tbody>
</table>

### Table 3: Patents on the liquid-crystal display

<table>
<thead>
<tr>
<th>Patent</th>
<th>Date</th>
<th>Author(s) and Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>4017416</td>
<td>Apr 1, 1978</td>
<td>Coates, et al. Liquid crystalline materials</td>
</tr>
<tr>
<td>4472592</td>
<td>Apr 4, 1986</td>
<td>Hsu. Liquid crystal composition and method</td>
</tr>
<tr>
<td>4514044</td>
<td>Apr 9, 1985</td>
<td>Eidenschink, et al. Liquid crystalline hexahydroterphenyl compounds</td>
</tr>
<tr>
<td>4682234</td>
<td>Apr 19, 1988</td>
<td>Hsu. Liquid crystal compositions and method</td>
</tr>
<tr>
<td>4793179</td>
<td>Apr 26, 1988</td>
<td>Petrzilka, et al. Novel liquid crystal materials and devices incorporating such compounds</td>
</tr>
<tr>
<td>4832462</td>
<td>May 13, 1988</td>
<td>Gunjima, et al. 2,2'-difluoro-4-alkoxy-4'-hydroxydiphenyls and their use in liquid crystal display devices</td>
</tr>
<tr>
<td>4915470</td>
<td>May 13, 1989</td>
<td>Hsu. Liquid crystal compositions and method</td>
</tr>
<tr>
<td>4930252</td>
<td>May 18, 1989</td>
<td>Hsu. Liquid crystal compositions and method</td>
</tr>
<tr>
<td>5016989</td>
<td>May 18, 1990</td>
<td>Hsu. Liquid crystal compositions and method</td>
</tr>
<tr>
<td>5227228</td>
<td>Jan 10, 1991</td>
<td>Gray, et al. Liquid crystal nematic compounds and their use in liquid crystal display devices</td>
</tr>
<tr>
<td>5351570</td>
<td>Jan 3, 1994</td>
<td>Varnier, et al. 2,2'-difluoro-4-alkoxy-4'-hydroxydiphenyls and their use in liquid crystal display devices</td>
</tr>
<tr>
<td>5400084</td>
<td>Apr 19, 1991</td>
<td>Lock, et al. Liquid crystal display devices</td>
</tr>
</tbody>
</table>

---

**Liquid crystal display**

[Graphical representation of liquid crystal display components]
Triangular connectivity and triangular networks

We can assign to a given graph a triangular network in which every line of the original graph gets as its weight the number of triangles that contain it. The triangular weights provide us, combined with islands, with a very efficient way to identify dense parts of a graph.

These notions can be generalized to short cycle connectivity (see paper).
Edge-cut at level 16 of triangular network of Erdős collaboration graph

without Erdős,

\( n = 6926, \quad m = 11343 \)
Edge-cut at level 11 of transitive network of ODLIS dictionary graph
Islands – The Edinburgh Associative Thesaurus

\[ n = 23219, \ m = 325624, \ \text{transitivity weight} \]
Pattern searching

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 molecule (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.
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
Relinking patterns in p-graphs

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

- second character: number of vertices 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 vertices equals to the number of last vertices.
Frequencies normalized with number of couples in p-graph $\times 1000$.

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

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
KEDS

Standard approach:

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

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

We decided to merge actions into 3 groups

Positive (blue)       Neutral (green)       Negative (red)

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
KEDS statistics

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

```r
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")
```
Temporal Analysis of US Patents Network


http://www.nber.org/patents/

- developed between 1975 – 1999
granted patents from January 1963 – December 1999
- 2923922 patents with text descriptions, 850846 as image
  3774768 vertices
- 16522438 citations (network arcs)

Several variables (properties of vertices) are also available: application year, assignee identifier, technological (sub)category, …
Shrinking of network according to categories & time slices

All vertices from the same category in the same time slice are \textit{shrunken} in one vertex.

The obtained smaller networks over time are analyzed.

For looking closer to a specific segment of the network subcategories or assignee numbers can be used.
Choice of sliding time window

We used the knowledge about *backward citation lags* (Hall, Jaffe, Trajtenberg), that is the time difference between grant year of the citing patent and that of the cited patents. The highest number of cited patents were granted 3 and 4 years earlier. For even older patents the number drastically decreases.

Since application year and grant year somehow correlate, we used time slices of **4 years with no history**. All the citations lagged more than 4 years were excluded.

Possible interpretation:

- less lagged citations could be part of the research and development at current time
- other citations used as references to well known methods patented earlier
Growth of number of patents and relative growth of citations within category

Number of patents in 4 years time slices

Relative growth of citations in 4 years time slices
Development of technological categories

How do connections among categories develop?

We used the *hubs and authorities* procedure implemented in *Pajek*.

Vertex is a *good hub* if it points to many good authorities, and it is a *good authority* if it is pointed by many good hubs.

Every vertex $v$ in the network gets two weights ($x_v$ and $y_v$). The corresponding vectors $x$ and $y$ are the principal eigenvectors of matrices $W^T W$ and $WW^T$, where $W$ is the adjacency matrix of the network.
Hubs

Categories with large values of hubs (Computers & Communication and Mechanical, Others) are categories which combine knowledge from other important technological categories.
Categories with large values of authorities (Mechanical and Electrical & Electronic) play very important role in setting the foundations – \textit{basic knowledge}.
PajektoSVGanim

For movie-like ’smooth’ visualization of evolution of a network through time we are developing a separate program PajektoSVGanim (implemented by Darko Brvar) that transforms a sequence of Pajek’s layouts into a SVG animation.

Plans: 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.
Further Readings / Books


Further Readings / Papers


Some links

- *Pajek Data Sets*. [Page](http://vlado.fmf.uni-lj.si/pub/networks/pajek/data/).
- *The InfoVis CyberInfrastructure Software*. [Page](http://infovis.pbrc.org/).
- *Last version of these slides*. [Page](http://www.kdd2005.org/).