Pajek
Software tool for SNA

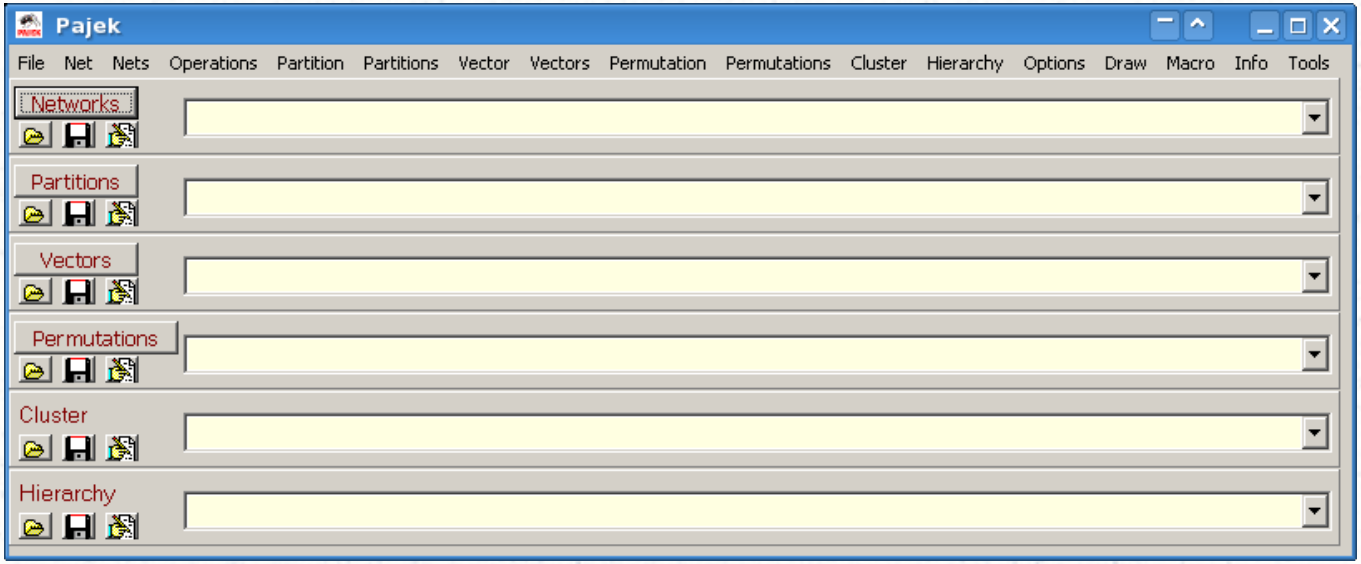
Jožef Stefan International Postgraduate school

Vid Podpečan, Jure Ferlež
{vid.podpecan, jure.ferlez} @ ijs.si

Intro

- software for network analysis
- authors:
  - V. Batagelj, A. Mrvar (University of Ljubljana)
- free for non-commercial use
- works on Windows
- on Linux: with Wine emulator

Interface



- decomposition of networks
- visualization
- algorithms for analysis (shortest paths, flows, ...)
- find, extract, analyze and show subgraphs
- ...

Pajek data types

- 6 data types:
  - **network:** graph
  - **partition:** nominal properties of vertices
  - **vector:** numerical properties of vertices
  - **cluster:** subset of vertices
  - **permutation:** reordering of vertices
  - **hierarchy:** tree structure on vertices
- transformation between data structures
- stack history of all data

Pajek menus: how to

- commands that need only a network:
  - menu Net
- commands that need two networks:
  - menu Nets
- commands that need two objects:
  - menu Operations
- commands that need only a partition/vector/..
  - menu Partition / Vector / Permutation /...

Loading data

- many graph data types:
  - GEDCOM, UCINET, MDL MOL, ...
  - native format: **.net**
- **.vec**, **.clu**, **.per** files for other data
- project file which stores everything
- does not support Unix text files
  - all lines must end with CR LF (Windows style)

Input files

inter.net

```
*Vertices  20
  1 "m01"
  2 "m02"
  3 "m03"
  4 "m04"
  5 "m05"
  6 "f06"
  7 "f07"
  8 "f08"
  9 "f09"
 10 "f10"
 11 "f11"
 12 "f12"
 13 "f13"
 14 "f14"
 15 "f15"
 16 "f16"
 17 "f17"
 18 "f18"
 19 "f19"
 20 "f20"
*Edges
  1   2   2
  1   3  10
  1   4   4
```

inter.net

```
  1   6   5
  1   7   9
  1   8   7
  1   9   4
  1  10   3
  1  11   3
  1  12   7
  1  13   3
  1  14   2
  1  15   5
  1  16   1
  1  17   4
  1  18   1
  2   3   5
  2   4   1
  2   5   3
  2   6   1
  2   7   4
  2   8   2
  2   9   6
  2  10   2
  2  11   5
  2  12   4
  2  13   3
  2  14   2

...
```

sex.clu

```
*vertices 20
1
1
1
1
1
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
```

age.vec

```
*vertices 20
15
10
10
 8
 7
15
 5
11
 8
 9
16
10
14
 5
 7
11
 7
 5
15
 4
```

rank.per

```
*vertices 20
 1
 2
 3
 4
 5
10
11
 6
12
 9
 7
 8
18
19
20
13
14
15
16
17
```

Input files

- .net files: three blocks:
  - *Vertices
  - *Arcs or *Arclist
  - *Edges or *Edgeslist
- various parameters:
  - name, color, weight, shape, position,...
- more info: Pajek manual
  - http://vlado.fmf.uni-lj.si/pub/networks/pajek/doc/pajekman.pdf

Loading and drawing graphs

- data: as1.net (relations in a class)
  - who would you invite to a party ?
- load:
  - File – Network – Read
- draw:
  - Draw – Draw
  - Layout – Energy – Kamada-Kawai – Free
- zoom:
  - click and drag right mouse button

Visual properties

- draw window:
  - button GraphOnly: hide all but graph
- meni Options:
  - label
  - color
  - size
  - width
- previous / next graph on the stack

Partitions

- load partition as1.clu (male / female):
  - File – Partition – Read
  - another possibility: partitions from shapes:
    - Net – Partitions – Vertex shapes
- draw partitions:
  - Draw – Draw partition
  - each partition can be moved separately
- extract both subnetworks, defined by partitions:
  - Operations – Extract from network – Partition
  - enter number 1 (female), repeat with 2 (male)

Partitions and reduction: world trade data

- load project file SaK_trade.paj:
  - File – Pajek project file – Read
  - Info – Partition
- shrink all partitions:
  - Operations – Shrink network – Partition
    - first value: 1 (default)
    - second value: 0 (shrink all)
- now shrink all except South America
  - second value = 4
- remove lines inside South America:
  - Operations – Transform – Remove lines – Inside clusters

Shortest paths

- load dic28.net (english dictionary):
  - relation: one changed character
- find shortest path between two words:
  - e.g. “black” and “white”
  - Net – Paths between 2 vertices – One shortest
  - Draw
- repeat, now find all shortest paths:
  - Net – Paths between 2 vertices – All shortest
  - Draw

Weak and strong components

- load as10.net:
  - whom would you select to inform you about important news in school ?
- find strong components:
  - Net – Components – Strong components
- Draw – Draw partition
- comment strong componets (male / female population)
- how many weak components ?
- Shrink:
  - Operation - Shrink network - Partition

Degree centrality

- load as10.net
- input degree partitions:
  - Net – Partitions – Degree – Input
- Draw - Draw partition
- we also have a vector:
  - multiply by 15.0:
    - Vector – Transform – Multiply by
  - Draw – Draw vector
- which vertex is the most popular ?
  - Info - Partition

Extracting important parts

- load USAir94.net:
  - flight connections between US airports
- this graph has weighted lines:
  - remove all lines with weight <= 0.15
  - Net – Transform – Remove – Lines with value – Lower than
- now there are isolated vertices:
  - compute degree partition: Net – Partitions – Degree
  - Info – Partition
  - Operations – Extract from network – Partition
  - Draw
  - Options – Lines - Grayscale

Betweenness and closeness centrality

- load USAir94.net:
  - flight connections between US airports
- compute betweeness centrality:
  - Net – Vector – Centrality – Betweenness
  - multiply vector by 15.0
- Draw – Draw vector
- which airports are the most important?
- repeat the procedure for closeness centrality

Neighbourhood

- load USAir94.net:
- compute k-neighbourhood of an important airport (e.g. 118 - Chicago O'Hare)
  - Net – k-Neighbours
- inspect results:
  - Info – Partition
- explain results:
  - how many can be reached in 2 flights ?
  - which is the farthest ?

Cores

- load as1.net
- find core hierarchy:
  - Net – Partition – Core – Input
  - Draw – Draw Partition
  - Info - Partition
- extract the strongest and weakest core:
  - Operations – Extract from network – Partition
  - Draw – Draw partition
- what can you say about weakest / strongest core?
- explain core hierarchy

Islands

- load eatRS.net:
  - Edinburgh associative thesaurus
- lines are weighted: we can find line islands:
  - Net – Partitions – Islands – Line weights
  - min=5, max=15
- extract partition (island) number 187
  - Operations – Extract from network – Partition
- draw this island

Two mode networks: cores

- load TheyRuleR.net
  - network of companies and directors
- get 2-mode partitions:
  - Net – Partitions – 2-mode
- draw window:
  - Layers – In y direction
- get border (p,q) core values:
  - Net – Partitions – Core – 2-Mode border

Two mode networks: cores

- inspect report and find core (1, 52)
  - Net – Partition – Core -2-Mode
  - p=1, q=52
- extract partition 1:
  - Operations – Extract from network – Partition
- get 2-mode partitions:
  - Net – Partitions – 2-mode
- Draw – Draw partition
- Spin – Normal: 0 0 0
- Spin – Spin around: 90 degrees

Two mode networks

- explain (1, 52) core
- repeat the process for the (8,1) core
- convert the network into 1-mode:
  - values on lines or multiple lines
  - loop or no loops
  - row or columns (first or second set)
  - Net – Transform – 2-Mode to 1-Mode
- now standard methods can be used

Hierarchical clustering

- load as1.net
- first, create one cluster with all vertices:
  - Cluster – Create complete cluster
- then compute dissimilarity matrix:
  - Operations – Dissimilarity – Network based – Corrected Euclidean
  - save dendrogram into a file (.eps)
- inspect created dendrogram:
  - how many clusters seem appropriate?

Hierarchical clustering

- open hierarchy window:
  - 

- expand the tree until it seems ok to you
- close each terminal branch:
  - click on brach
  - press Ctrl-T until (Close) appears
- create partition from hierarchy:
  - Hierarchy – Make Partition
- draw partitions
- shrink partitions and draw (Operations – Shrink - Partition)