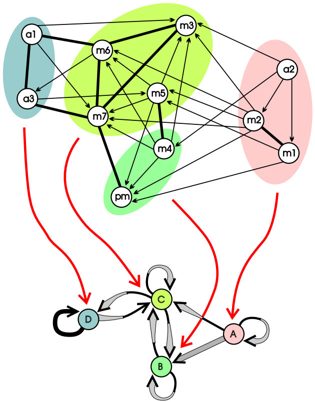
Analiza omrežij

9. Razvrstitve in bločni modeli

Vladimir Batagelj

Magistrski program Uporabna statistika
Ljubljana, maj 2024

Kazalo

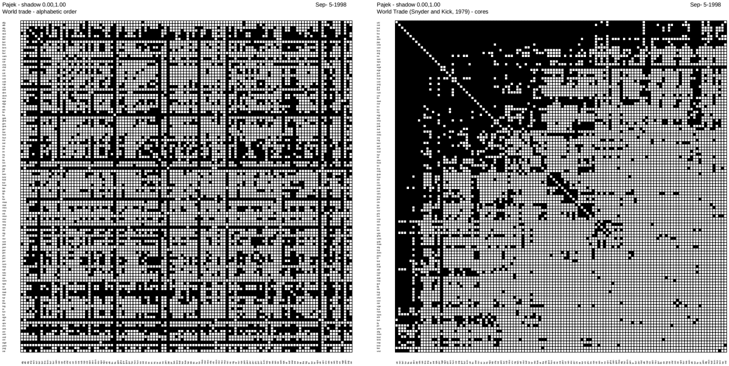


prof. Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
prosojnice (PDF)

20. maj 2024 ob 18 : 02/ maj 2013

Preurejanja in bločni modeli

Snyder & Kickovo omrežje svetovne trgovine / $n = 118$, $m = 514$



Države urejene po abecedi (levo) urejenost iz razvrstitve (desno)

Preurejanje matrik

Dano matriko lahko preuredimo – določimo *urejenost* ali *permutacijo* njenih vrstic in stolpcev – na več načinov. Prikaz ustrezno preurejene matrike nam omogoča razkriti njeno zgradbo. Nekaj pristopov:

- urejenost po izbrani lastnosti: po šibkih/krepkih komponentah povezanosti; po stopnjah; po središčnih številih; po komponentah v središčnih slojih;
- urejenost glede na hierarhično razvrstitev.

Obstajajo tudi posebni postopki za preurejanje matrik: površčanje in kupčkanje (Murtagh), Cuthill-McKee-jev postopek, . . .

```
Partition/Make Permutation
Vector/Make Permutation
Network/Create Permutation/Reverse Cuthill-McKee
Network/Create Permutation/Core+Degree
```

Površčanje (seriation)

Naj bo dano dvovrstno omrežje $\mathcal{N} = (\mathcal{U}, \mathcal{W}, R, w)$, $R \subseteq \mathcal{U} \times \mathcal{W}$, $w : R \to \mathbb{R}_0^+$.
Preslikavi $\rho : \mathcal{U} \to 1..n$, $\sigma : \mathcal{W} \to 1..m$ naj bosta bijekciji (permutaciji).
Za vrstično enoto $\mathrm{X} \in \mathcal{U}$ vpeljemo vrstično vsoto $r(\mathrm{X})$; za stolpčno enoto $\mathrm{Y} \in \mathcal{W}$ pa stolpčno vsoto $c(\mathrm{Y})$

$$r(\mathrm{X}) = \sum_{\mathrm{Y} \in R(\mathrm{X})} w(\mathrm{X}, \mathrm{Y}) \quad \text{in} \quad c(\mathrm{Y}) = \sum_{\mathrm{X} \in R^{-1}(\mathrm{Y})} w(\mathrm{X}, \mathrm{Y})$$

ter naprej vrstične uteži $p(\mathrm{X})$ in stolpčne uteži $q(\mathrm{Y})$

$$p(\mathrm{X}) = \frac{1}{r(\mathrm{X})} \sum_{\mathrm{Y} \in R(\mathrm{X})} \sigma(\mathrm{Y}) w(\mathrm{X}, \mathrm{Y}), \quad q(\mathrm{Y}) = \frac{1}{c(\mathrm{Y})} \sum_{\mathrm{X} \in R^{-1}(\mathrm{Y})} \rho(\mathrm{X}) w(\mathrm{X}, \mathrm{Y})$$

Če je $r(\mathrm{X}) = 0$, je tudi $p(\mathrm{X}) = 0$; če je $c(\mathrm{X}) = 0$, je $q(\mathrm{X}) = 0$.

. . . postopek površčanja

določi (slučajno) σ ;
repeat
  določi vrstične uteži $p(\mathrm{X})$, $\mathrm{X} \in \mathcal{U}$; $\rho :=$ sort_decreasing$(\mathcal{U}, p)$;
  določi stolpčne uteži $q(\mathrm{Y})$, $\mathrm{Y} \in \mathcal{W}$; $\sigma :=$ sort_decreasing$(\mathcal{W}, q)$;
until urejenosti se ustalita (ali število korakov doseže mejo)

Za enovrstna omrežja je $\rho = \sigma$ in za utež enote uporabimo $p(\mathrm{X}) + q(\mathrm{X})$.

`Operations/Network+Permutation/Numbering*/Seriation`

Kupčkanje (clumping)

Kupčkanje poskuša preurediti enote tako, da mera *nakopičenosti* doseže čim večjo vrednost. Urejenost se določa postopno s požrešnim dodajanjem nove enote na najustreznejše mesto v tekoči urejenosti določeni s seznamom, ki vsebuje k enot in na obeh koncih še dodatni 'stražarski' enoti $[\mathrm{X}_0 = \mathbf{0}, \mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_k, \mathbf{0} = \mathrm{X}_{k+1}]$.
Če vstavimo vrstično enoto X v seznam za členom X_i, to ustvari vrstično nakopičenost

$$Q(i) = \sum_{\mathrm{Y} \in R(\mathrm{X}) \cap (R(\mathrm{X}_i) \cup R(\mathrm{X}_{i+1}))} w(\mathrm{X},\mathrm{Y})(w(\mathrm{X}_i,\mathrm{Y}) + w(\mathrm{X}_{i+1},\mathrm{Y}))$$

Če $(\mathrm{X},\mathrm{Y}) \notin R$, je $w(\mathrm{X},\mathrm{Y}) = 0$.
Na enak način (obratno omrežje) lahko preuredimo tudi stolpčne enote.
Za enovrstna omrežja uporabimo sestavljeno mero – vsoto vrstične in stolpčne nakopičenosti.

. . . postopek kupčkanja

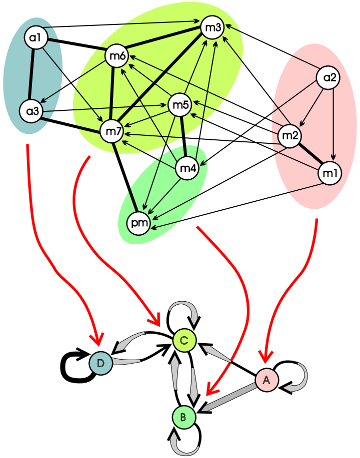
izberi (slučajno) $\mathrm{X} \in \mathcal{U}$; $\mathbf{S} := \mathcal{U} \setminus \{\mathrm{X}\}$
order $:= [\mathbf{0}, \mathrm{X}, \mathbf{0}]$; $k := 1$;
while $\mathbf{S} \neq \emptyset$ **do begin**
 izberi $\mathrm{X} \in \mathbf{S}$; $\mathbf{S} := \mathbf{S} \setminus \{\mathrm{X}\}$;
 for $i := 0$ **to** k **do** določi $Q(i)$;
 $j := \mathrm{argmax}_i\, Q(i)$;
 vstavi enoto X v urejenost *order* za enoto X_j;
 $k := k + 1$
end;

`Operations/Network+Permutation/Numbering*/Clumping`

Bločni modeli kot problem razvrščanja

Cilj *bločnega modeliranja* je skrčiti obsežno, morda neskladno omrežje v manjše, razumljivejše omrežje, ki ustrezno povzema zgradbo izvornega omrežja in ponuja njeno razlago.

Enote/vozlišča omrežja so pri bločnem modeliranju razvrščene v skupine glede na neko *smiselno* enakovrednost.



Skupina, razvrstitev, blok

Glavni cilj bločnega modeliranja je določiti v danem omrežju $\mathcal{N} = (\mathcal{U}, R)$, $R \subseteq \mathcal{U} \times \mathcal{U}$, *skupine* (razrede) enot, ki imajo enake ali podobne strukturne značilnosti glede na R – enote iz posamezne skupine so enako ali podobno povezane z enotami drugih skupin.
Skupine sestavljajo *razvrstitev* $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ ki je *razbitje* množice enot $\mathcal{U}$.
Kot vemo, vsako razbitje določa neko enakovrednost (in obratno).
Označimo s $\sim$ enakovrednost določeno z razbitjem $\mathbf{C}$.
Razvrstitev $\mathbf{C}$ razbije tudi relacijo R na *bloke*

$$R(C_i, C_j) = R \cap C_i \times C_j$$

Vsak tak blok sestavljajo enote iz skupin C_i in C_j in vse povezave, ki vodijo iz skupine C_i v skupino C_j. Če je $i = j$, bloku $R(C_i, C_i)$ rečemo *diagonalni* blok.

Strukturna in regularna enakovrednost

Večina vrst enakovrednosti temelji na ne enem od naslednjih dveh pristopov/pogledov (Faust, 1988):

- enakovredni enoti sta na enak način povezani (imata enak vzorec povezanosti) do **istih** sosedov;
- enakovredni enoti sta na enak ali podoben način povezani do (lahko) **različnih** sosedov.

Primer prve vrste enakovrednostni je strukturna enakovrednost; primer druge vrste pa regularna enakovrednost.

Strukturna enakovrednost

Enoti sta enakovredni, če sta povezani z ostalim omrežjem na *enak* način (Lorrain in White, 1971). Natančneje:
Enoti X in Y sta *strukturno enakovredni*, kar zapišemo $X \equiv Y$, ntk. ki je permutacija (premena) $\pi = (X\,Y)$ avtomorfizem of relacije R (Borgatti in Everett, 1992).
Drugače povedano, enoti X in Y sta strukturno enakovredni ntk.:

s1.	$XRY \Leftrightarrow YRX$	s3.	$\forall Z \in \mathcal{U} \setminus \{X, Y\} : (XRZ \Leftrightarrow YRZ)$
s2.	$XRX \Leftrightarrow YRY$	s4.	$\forall Z \in \mathcal{U} \setminus \{X, Y\} : (ZRX \Leftrightarrow ZRY)$

...strukturna enakovrednost

Za strukturno enakovrednost sta edina mogoča bloka prazni in polni blok (z morebitnim preklopom na diagonali za diagonalne bloke):

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{matrix}$$

Regularna enakovrednost

V dejanskih omrežjih je le malo enot strukturno enakovrednih. Skupni imenovalec vsem poskusom njene posplošitve je, da sta enoti enakovredni, če sta podobno povezani z drugimi enakovrednimi enotami.
White in Reitz (1983): Enakovrednost $\approx$ on $\mathcal{U}$ je *regularna enakovrednost* na omrežju $\mathcal{N} = (\mathcal{U}, R)$ ntk. za vse enote $\mathrm{X}, \mathrm{Y}, \mathrm{Z} \in \mathcal{U}$, iz $\mathrm{X} \approx \mathrm{Y}$ izhaja tudi

$$\begin{array}{ll} \text{R1.} & \mathrm{X}R\mathrm{Z} \Rightarrow \exists \mathrm{W} \in \mathcal{U} : (\mathrm{Y}R\mathrm{W} \wedge \mathrm{W} \approx \mathrm{Z}) \\ \text{R2.} & \mathrm{Z}R\mathrm{X} \Rightarrow \exists \mathrm{W} \in \mathcal{U} : (\mathrm{W}R\mathrm{Y} \wedge \mathrm{W} \approx \mathrm{Z}) \end{array}$$

Drug pogled na regularno enakovrednost temelji na barvanjih vozlišč (Everett, Borgatti 1996). Naj bo $b : \mathcal{U} \to B$ barvanje vozlišč omrežja. Enakovrednost $\approx$ je *regularna* ntk. za vsak par $\mathrm{X}, \mathrm{Y} \in \mathcal{U}$ velja

$$\mathrm{X} \approx \mathrm{Y} \Leftrightarrow (b(\mathrm{X}) = b(\mathrm{Y})) \wedge (b(N(\mathrm{X})) = b(N(\mathrm{Y})))$$

... regularna enakovrednost

Theorem (Batagelj, Doreian, Ferligoj, 1992)

Naj bo $\mathbf{C} = \{C_i\}$ *razbitje, ki določa regularno enakovrednost* $\approx$ *na omrežju* $\mathcal{N} = (\mathcal{U}, R)$. *Tedaj je vsak blok* $R(C_u, C_v)$ *ali prazen ali pa ima lastnost da obstaja vsaj ena 1 v vsaki njegovi vrstici in vsakem njegovem stolpcu. Obratno, če so za dano razbitje* $\mathbf{C}$ *vsi bloki teh dveh vrst, je ustrezna enakovrednost regularna.*

Bloki regularne enakovrednosti so prazni ali 1-pokriti.

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{matrix}$$

Določanje bločnih modelov

The problem of establishing a partition of units in a network in terms of a selected type of equivalence is a special case of *clustering problem* that can be formulated as an optimization problem (Φ, P) as follows:

Determine the clustering $\mathbf{C}^\star \in \Phi$ *for which*

$$P(\mathbf{C}^\star) = \min_{\mathbf{C} \in \Phi} P(\mathbf{C})$$

where Φ is the set of *feasible clusterings* and P is a *criterion function*. Since the set of units $\mathcal{U}$ is finite, the set of feasible clusterings is also finite. Therefore the set $\text{Min}(\Phi, P)$ of all solutions of the problem (optimal clusterings) is not empty.

- *posredni* pristop – kriterijsko funkcijo izrazimo na osnovi mere različnosti med vozlišči;
- *premi* pristop – kriterijska funkcija meri odstopanje dejanske razvrstitve vozlišč omrežja od ustreznega idealnega stanja za

Posredni pristop

$$
\begin{tikzcd}
R \arrow[dd] \arrow[dr] & \\
 & Q \arrow[dl] \\
D \arrow[d] & \\
 &
\end{tikzcd}
$$

OMREŽJE — R

OPISI ENOT — Q: omrežna relacija, matrika poti, trojice, orbite

MATRIKA RAZLIČNOSTI — D

OBIČAJNI POSTOPKI RAZVRŠČANJA — postopki združevanja, postopki prestavljanj, metoda voditeljev, itd.

Dissimilarities

The property $t : \mathcal{U} \to \mathbb{R}$ is *structural property* if, for every automorphism φ, of the relation R, and every unit $x \in \mathcal{U}$, it holds that $t(x) = t(\varphi(x))$. Some examples of a structural property include

$t(u) =$ the *degree* of unit u;
$t(u) =$ number of units at *distance* d from the unit u;
$t(u) =$ number of *triad*s of type x at the unit u.

Centrality measures are further examples of structural properties. We can define the description of the unit u as $[u] = [t_1(u), t_2(u), \ldots, t_m(u)]$. As a simple example, t_1 could be *degree* centrality, t_2 could be *closeness* centrality and t_3 could be *betweenness* centrality. The dissimilarity between units u and v could be defined as $d(u, v) = D([u], [v])$ where D is some (standard) dissimilarity between real vectors. In the simple example, D could be the *Euclidean* distance between the centrality profiles.

Dissimilarities based on matrices

We consider the following list of dissimilarities between units x_i and x_j where the description of the unit consists of the row and column of the property matrix $\mathbf{Q} = [q_{ij}]$. We take as units the rows of the matrix

$$\mathbf{X} = [\mathbf{Q} \mid \mathbf{Q}^T]$$

. . . Dissimilarities

Manhattan distance: $d_m(x_i, x_j) = \sum_{s=1}^{n} (|q_{is} - q_{js}| + |q_{si} - q_{sj}|)$

Euclidean distance:

$$d_E(x_i, x_j) = \sqrt{\sum_{s=1}^{n} ((q_{is} - q_{js})^2 + (q_{si} - q_{sj})^2)}$$

Truncated Manhattan distance:

$$d_s(x_i, x_j) = \sum_{\substack{s=1 \\ s \neq i,j}}^{n} (|q_{is} - q_{js}| + |q_{si} - q_{sj}|)$$

Truncated Euclidean distance (Faust, 1988):

$$d_S(x_i, x_j) = \sqrt{\sum_{\substack{s=1 \\ s \neq i,j}}^{n} ((q_{is} - q_{js})^2 + (q_{si} - q_{sj})^2)}$$

... Dissimilarities

Corrected Manhattan-like dissimilarity ($p \geq 0$):

$$d_c(p)(x_i, x_j) = d_s(x_i, x_j) + p \cdot (|q_{ii} - q_{jj}| + |q_{ij} - q_{ji}|)$$

Corrected Euclidean-like dissimilarity (Burt and Minor, 1983):

$$d_e(p)(x_i, x_j) = \sqrt{d_S(x_i, x_j)^2 + p \cdot ((q_{ii} - q_{jj})^2 + (q_{ij} - q_{ji})^2)}$$

Corrected dissimilarity:

$$d_C(p)(x_i, x_j) = \sqrt{d_c(p)(x_i, x_j)}$$

The parameter, p, can take any positive value. Typically, $p = 1$ or $p = 2$, where these values count the number of times the corresponding diagonal pairs are counted.

`Operations/Network+Cluster/Dissimilarity*/Network based`

. . . Dissimilarities

It is easy to verify that all expressions from the list define a dissimilarity (i.e. that $d(x, y) \geq 0$; $d(x, x) = 0$; and $d(x, y) = d(y, x)$). Each of the dissimilarities from the list can be assessed to see whether or not it is also a distance: $d(x, y) = 0 \Rightarrow x = y$ and $d(x, y) + d(y, z) \geq d(x, z)$.
The dissimilarity measure d is *compatible* with a considered equivalence $\sim$ if for each pair of units holds

$$X_i \sim X_j \Leftrightarrow d(X_i, X_j) = 0$$

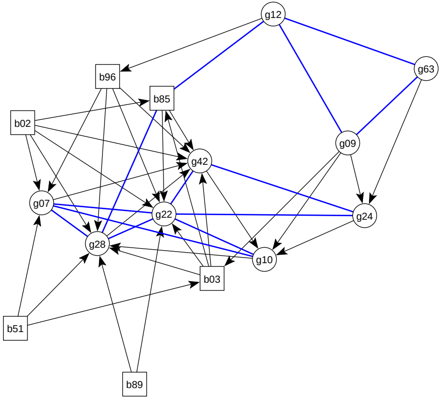
Not all dissimilarity measures typically used are compatible with structural equivalence. For example, the *corrected Euclidean-like dissimilarity* is compatible with structural equivalence.
The indirect clustering approach does not seem suitable for establishing clusterings in terms of regular equivalence since there is no evident way how to construct a compatible (dis)similarity measure.

Example: Support network among informatics students

The analyzed network consists of social support exchange relation among fifteen students of the Social Science Informatics fourth year class (2002/2003) at the Faculty of Social Sciences, University of Ljubljana. Interviews were conducted in October 2002.
Support relation among students was identified by the following question:

> *Introduction: You have done several exams since you are in the second class now. Students usually borrow studying material from their colleagues.*
>
> *Enumerate (list) the names of your colleagues that you have most often borrowed studying material from. (The number of listed persons is not limited.)*

Class network

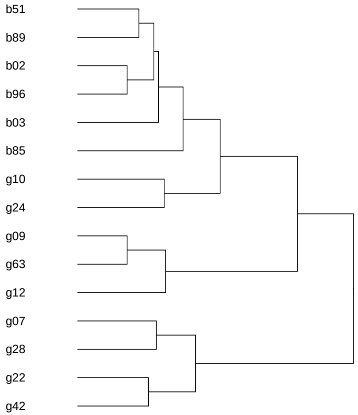


`class.net`

Vertices represent students in the class; circles – girls, squares – boys. Opposite pairs of arcs are replaced by edges.

```
Cluster/Create Complete Cluster
Operations/Network+Cluster/Dissimilarity*/Network based/
```

Indirect approach



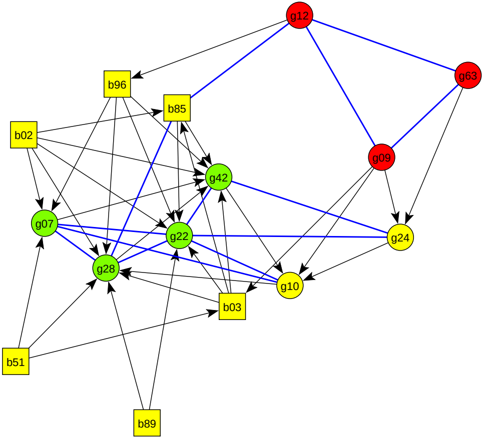
Using *Corrected Euclidean-like dissimilarity* and *Ward clustering method* we obtain the following dendrogram.

From it we can determine the number of clusters: 'Natural' clusterings correspond to clear 'jumps' in the dendrogram.

If we select 3 clusters we get the partition **C**.

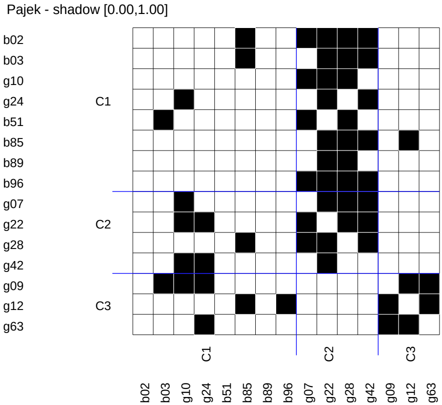
$$\mathbf{C} = \{\{b51, b89, b02, b96, b03, b85, g10, g24\}, \{g09, g63, g12\}, \{g07, g28, g22, g42\}\}$$

Partition in 3 clusters



On the picture, vertices in the same cluster are of the same color.

Matrix



The partition can be used also to reorder rows and columns of the matrix representing the network. Clusters are divided using blue vertical and horizontal lines.

```
determine clustering in the hierarchy
Hierarchy/Make Permutation
Hierarchy/Make Partition
File/Network/Export as Matrix to EPS/Using Permutation
```

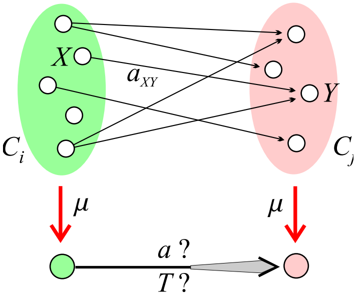
Direct Approach and Generalized Blockmodeling

The second possibility for solving the blockmodeling problem is to construct an appropriate criterion function directly and then use a local optimization algorithm to obtain a 'good' clustering solution.
Criterion function $P(\mathbf{C})$ has to be *sensitive* to considered equivalence:

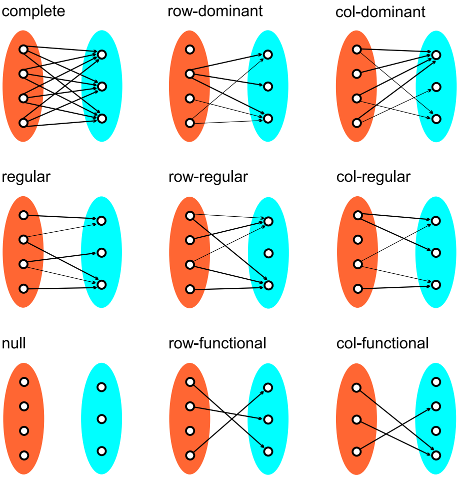
$$P(\mathbf{C}) = 0 \Leftrightarrow \mathbf{C} \text{ defines considered equivalence.}$$

Generalized Blockmodeling

A *blockmodel* consists of structures obtained by identifying all units from the same cluster of the clustering **C**. For an exact definition of a blockmodel we have to be precise also about which blocks produce an arc in the *reduced graph* and which do not, and of what *type*. Some types of connections are presented in the figure on the next slide. The reduced graph can be represented by relational matrix, called also *image matrix*.



Block Types


complete
row-dominant
col-dominant
regular
row-regular
col-regular
null
row-functional
col-functional

Generalized equivalence / Block Types

$$\begin{array}{c|ccccc|} & \multicolumn{5}{c|}{Y} \\ \hline & 1 & 1 & 1 & 1 & 1 \\ X & 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

complete

$$\begin{array}{c|ccccc|} & \multicolumn{5}{c|}{Y} \\ \hline & 0 & 1 & 0 & 0 & 0 \\ X & 1 & 1 & 1 & 1 & 1 \\ & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 1 & 0 \\ \hline \end{array}$$

row-dominant

$$\begin{array}{c|ccccc|} & \multicolumn{5}{c|}{Y} \\ \hline & 0 & 0 & 1 & 0 & 0 \\ X & 0 & 0 & 1 & 1 & 0 \\ & 1 & 1 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 & 1 \\ \hline \end{array}$$

col-dominant

$$\begin{array}{c|ccccc|} & \multicolumn{5}{c|}{Y} \\ \hline & 0 & 1 & 0 & 0 & 0 \\ X & 1 & 0 & 1 & 1 & 0 \\ & 0 & 0 & 1 & 0 & 1 \\ & 1 & 1 & 0 & 0 & 0 \\ \hline \end{array}$$

regular

$$\begin{array}{c|ccccc|} & \multicolumn{5}{c|}{Y} \\ \hline & 0 & 1 & 0 & 0 & 0 \\ X & 0 & 1 & 1 & 0 & 0 \\ & 1 & 0 & 1 & 0 & 0 \\ & 0 & 1 & 0 & 0 & 1 \\ \hline \end{array}$$

row-regular

$$\begin{array}{c|ccccc|} & \multicolumn{5}{c|}{Y} \\ \hline & 0 & 1 & 0 & 1 & 0 \\ X & 1 & 0 & 1 & 0 & 0 \\ & 1 & 1 & 0 & 1 & 1 \\ & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

col-regular

$$\begin{array}{c|ccccc|} & \multicolumn{5}{c|}{Y} \\ \hline & 0 & 0 & 0 & 0 & 0 \\ X & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

null

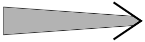
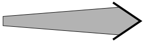
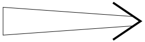
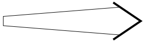
$$\begin{array}{c|ccccc|} & \multicolumn{5}{c|}{Y} \\ \hline & 0 & 0 & 0 & 1 & 0 \\ X & 0 & 0 & 1 & 0 & 0 \\ & 1 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 1 & 0 \\ \hline \end{array}$$

row-functional

$$\begin{array}{c|cccc|} & \multicolumn{4}{c|}{Y} \\ \hline & 1 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 0 \\ X & 0 & 0 & 1 & 0 \\ & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

col-functional

Characterizations of Types of Blocks

null	nul	all 0 *	
complete	com	all 1 *	
regular	reg	1-covered rows and columns	
row-regular	rre	each row is 1-covered	
col-regular	cre	each column is 1 -covered	
row-dominant	rdo	∃ all 1 row *	
col-dominant	cdo	∃ all 1 column *	
row-functional	rfn	∃! one 1 in each row	
col-functional	cfn	∃! one 1 in each column	
non-null	one	∃ at least one 1	

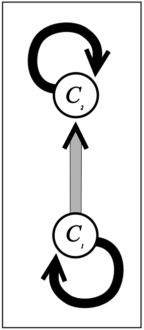
* except this may be diagonal

A block is *symmetric* iff $\forall X, Y \in C_i \times C_j : (XRY \Leftrightarrow YRX)$

Block Types and Matrices

$$
\begin{array}{cccc|cccc}
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}
$$

	C_1	C_2
C_1	complete	regular
C_2	null	complete



Formalization of blockmodeling

Let V be a set of positions or images of clusters of units. Let $\mu : \mathcal{U} \to V$ denote a mapping which maps each unit to its position. The cluster of units $C(t)$ with the same position $t \in V$ is

$$C(t) = \mu^{-1}(t) = \{X \in \mathcal{U} : \mu(X) = t\}$$

Therefore

$$\mathbf{C}(\mu) = \{C(t) : t \in V\}$$

is a partition (clustering) of the set of units $\mathcal{U}$.

Blockmodel

A *blockmodel* is an ordered sextuple $\mathcal{M} = (V, K, \mathcal{T}, Q, \pi, \alpha)$ where:

- V is a set of *types of units* (images or representatives of classes);
- $K \subseteq V \times V$ is a set of *connections*;
- $\mathcal{T}$ is a set of predicates used to describe the *types of connections* between different classes (clusters, groups, types of units) in a network. We assume that $\text{nul} \in \mathcal{T}$. A mapping $\pi : K \to \mathcal{T} \setminus \{\text{nul}\}$ assigns predicates to connections;
- Q is a set of *averaging rules*. A mapping $\alpha : K \to Q$ determines rules for computing values of connections.

A (surjective) mapping $\mu : \mathcal{U} \to V$ determines a blockmodel $\mathcal{M}$ of network $\mathcal{N}$ iff it satisfies the conditions: $\forall (t, w) \in K : \pi(t, w)(C(t), C(w))$ and $\forall (t, w) \in V \times V \setminus K : \text{nul}(C(t), C(w))$.

Equivalences

Let $\sim$ be an equivalence relation over $\mathcal{U}$ and $[X] = \{Y \in \mathcal{U} : X \sim Y\}$. We say that $\sim$ is *compatible* with $\mathcal{T}$ over a network $\mathcal{N}$ iff

$$\forall X, Y \in \mathcal{U} \exists T \in \mathcal{T} : T([X], [Y]).$$

It is easy to verify that the notion of compatibility for $\mathcal{T} = \{\mathrm{nul}, \mathrm{reg}\}$ reduces to the usual definition of regular equivalence (White and Reitz 1983). Similarly, compatibility for $\mathcal{T} = \{\mathrm{nul}, \mathrm{com}\}$ reduces to structural equivalence (Lorrain and White 1971).
For a compatible equivalence $\sim$ the mapping $\mu \colon X \mapsto [X]$ determines a blockmodel with $V = \mathcal{U}/\sim$.
The problem of establishing a partition of units in a network in terms of a selected type of equivalence is a special case of **clustering problem** that can be formulated as an optimization problem.

Criterion function

One of the possible ways of constructing a criterion function that directly reflects the considered equivalence is to measure the fit of a clustering to an ideal one with perfect relations within each cluster and between clusters according to the considered equivalence.
Given a clustering $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$, let $\mathcal{B}(C_u, C_v)$ denote the set of all ideal blocks corresponding to block $R(C_u, C_v)$. Then the global error of clustering $\mathbf{C}$ can be expressed as

$$P(\mathbf{C}) = \sum_{C_u, C_v \in \mathbf{C}} \min_{B \in \mathcal{B}(C_u, C_v)} d(R(C_u, C_v), B)$$

where the term $d(R(C_u, C_v), B)$ measures the difference (error) between the block $R(C_u, C_v)$ and the ideal block B. d is constructed on the basis of characterizations of types of blocks. The function d has to be compatible with the selected type of equivalence.

. . . criterion function

For example, for structural equivalence, the term $d(R(C_u, C_v), B)$ can be expressed, for non-diagonal blocks, as

$$d(R(C_u, C_v), B) = \sum_{\mathrm{X} \in C_u, \mathrm{Y} \in C_v} |r_{\mathrm{XY}} - b_{\mathrm{XY}}|.$$

where r_{XY} is the observed tie and b_{XY} is the corresponding value in an ideal block. This criterion function counts the number of 1s in erstwhile null blocks and the number of 0s in otherwise complete blocks. These two types of inconsistencies can be weighted differently.
Determining the block error, we also determine the type of the best fitting ideal block (the types are ordered).
The criterion function $P(\mathbf{C})$ is *sensitive* iff $P(\mathbf{C}) = 0 \Leftrightarrow \mu$ (determined by $\mathbf{C}$) is an exact blockmodeling. For all presented block types sensitive criterion functions can be constructed (Batagelj, 1997).

Deviations Measures for Types of Blocks

We can efficiently test whether a block $R(X, Y)$ is of the type T by making use of the characterizations of block types. On this basis we can construct the corresponding deviation measures. The quantities used in the expressions for deviations have the following meaning:

s_t	– total block sum = # of 1s in a block,
n_r	= card(X) – # of rows in a block,
n_c	= card(Y)– # of columns in a block,
p_r	– # of non-null rows in a block,
p_c	– # of non-null columns in a block,
m_r	– maximal row-sum,
m_c	– maximal column-sum,
s_d	– diagonal block sum = # of 1s on a diagonal,
d	– diagonal error = $\min(s_d, n_r - s_d)$.

Throughout the number of elements in a block is $n_r n_c$.

. . . Deviations Measures for Types of Blocks

Connection	$\delta(X, Y; T)$	
null	s_t	nondiagonal
	$s_t + d - s_d$	diagonal
complete	$n_r n_c - s_t$	nondiagonal
	$n_r n_c - s_t + d + s_d - n_r$	diagonal
row-dominant	$(n_c - m_r - 1)n_r$	diagonal, $s_d = 0$
	$(n_c - m_r)n_r$	otherwise
col-dominant	$(n_r - m_c - 1)n_c$	diagonal, $s_d = 0$
	$(n_r - m_c)n_c$	otherwise
row-regular	$(n_r - p_r)n_c$	
col-regular	$(n_c - p_c)n_r$	
regular	$(n_c - p_c)n_r + (n_r - p_r)p_c$	
row-functional	$s_t - p_r + (n_r - p_r)n_c$	
col-functional	$s_t - p_c + (n_c - p_c)n_r$	
density γ	$\max(0, \gamma n_r n_c - s_t)$	

For the null, complete, row-dominant and column-dominant blocks it is necessary to distinguish diagonal blocks and non-diagonal blocks.

Solving the blockmodeling problem

The obtained optimization problem can be solved by local optimization. Once a partitioning μ and types of connection π are determined, we can also compute the values of connections by using averaging rules.

Benefits from Optimization Approach

- *ordinary / inductive blockmodeling*: Given a network $\mathcal{N}$ and set of types of connection $\mathcal{T}$, determine the model $\mathcal{M}$;
- *evaluation of the quality of a model, comparing different models, analyzing the evolution of a network* (Sampson data, Doreian and Mrvar 1996): Given a network $\mathcal{N}$, a model $\mathcal{M}$, and blockmodeling μ, compute the corresponding criterion function;
- *model fitting / deductive blockmodeling*: Given a network $\mathcal{N}$, set of types $\mathcal{T}$, and a family of models, determine μ which minimizes the criterion function (Batagelj, Ferligoj, Doreian, 1998).
- we can fit the network to a partial model and analyze the residual afterward;
- we can also introduce different constraints on the model, for example: units X and Y are of the same type; or, types of units X and Y are not connected; ...

Pre-specified blockmodeling

In the previous slides the inductive approaches for establishing blockmodels for a set of social relations defined over a set of units were discussed. Some form of equivalence is specified and clusterings are sought that are consistent with a specified equivalence.
Another view of blockmodeling is deductive in the sense of starting with a blockmodel that is specified in terms of substance prior to an analysis.
In this case given a network, set of types of ideal blocks, and a reduced model, a solution (a clustering) can be determined which minimizes the criterion function.

Pre-Specified Blockmodels

The pre-specified blockmodeling starts with a blockmodel specified, in terms of substance, **prior to an analysis**. Given a network, a set of ideal blocks is selected, a family of reduced models is formulated, and partitions are established by minimizing the criterion function.
The basic types of models are:

$$\begin{array}{|c|c|}\hline * & * \\ \hline * & 0 \\ \hline\end{array} \quad \begin{array}{|c|c|}\hline * & 0 \\ \hline * & * \\ \hline\end{array} \quad \begin{array}{|c|c|}\hline * & 0 \\ \hline 0 & * \\ \hline\end{array} \quad \begin{array}{|c|c|}\hline 0 & * \\ \hline * & 0 \\ \hline\end{array}$$

center - periphery, hierarchy, clustering, bipartition

Prespecified blockmodeling example

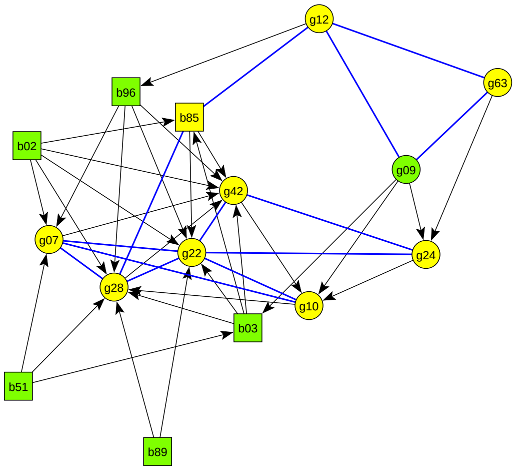
We expect that center-periphery model exists in the network: some students having good studying material, some not.
Prespecified blockmodel: (com/complete, reg/regular, -/null block)

	1	2
1	[com reg]	-
2	[com reg]	-

Using local optimization we get the partition:

$$\mathbf{C} = \{\{b02, b03, b51, b85, b89, b96, g09\},$$
$$\{g07, g10, g12, g22, g24, g28, g42, g63\}\}$$

2 clusters solution



Model

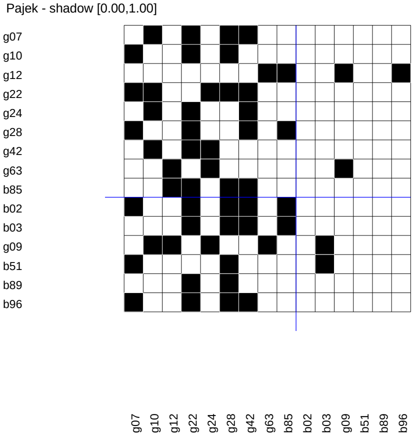


Image and Error Matrices:

	1	2
1	reg	-
2	reg	-

	1	2
1	0	3
2	0	2

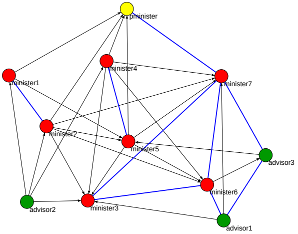
Total error $= 5$

The Student Government at the University of Ljubljana in 1992

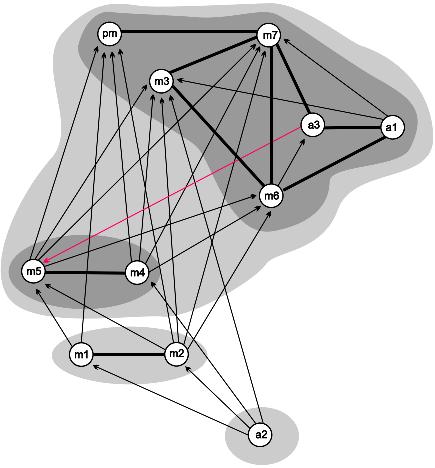
The relation is determined by the following question (Hlebec, 1993):

Of the members and advisors of the Student Government, whom do you most often talk with about the matters of the Student Government?

		m 1	p 2	m 3	m 4	m 5	m 6	m 7	m 8	a 9	a 10	a 11
minister 1	1	·	1	1	·	·	1	·	·	·	·	·
p.minister	2	·	·	·	·	·	·	·	1	·	·	·
minister 2	3	1	1	·	1	·	1	1	1	·	·	·
minister 3	4	·	·	·	·	·	·	1	1	·	·	·
minister 4	5	·	1	·	1	·	1	1	1	·	·	·
minister 5	6	·	1	·	1	1	·	1	1	·	·	·
minister 6	7	·	·	·	1	·	·	·	1	1	·	1
minister 7	8	·	1	·	1	·	·	1	·	·	·	1
adviser 1	9	·	·	·	1	·	·	1	1	·	·	1
adviser 2	10	1	·	1	1	1	·	·	·	·	·	·
adviser 3	11	·	·	·	·	·	1	·	1	1	·	·



A Symmetric Acyclic Blockmodel of Student Government

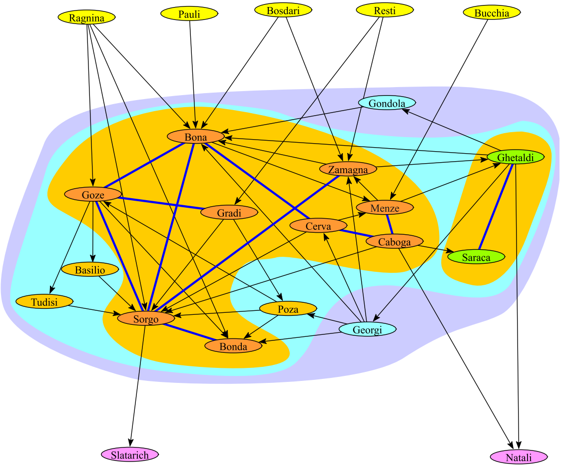


The obtained clustering in 4 clusters is almost exact. The only error is produced by the arc $(a3, m5)$.

Ragusan Noble Families Marriage Network, 18th and 19th Century

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Basilio	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.
Bona	2	.	.	.	.	.	.	1	.	.	.	2	.	2	.	.	.	.	.	.	.	1	.	.
Bonda	3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2	.	.
Bosdari	4	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1
Bucchia	5	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.
Caboga	6	.	.	.	.	.	.	1	.	.	.	.	.	1	1	.	.	.	.	1	.	1	.	.
Cerva	7	.	1	.	.	.	1	.	.	.	.	.	.	1	.	.	.	.	.	.	.	1	.	.
Georgi	8	.	1	2	.	.	.	1	.	.	.	.	.	.	.	.	4	.	.	.	.	.	.	1
Ghetaldi	9	.	1	.	.	.	.	.	1	.	1	.	.	.	1	.	.	.	.	1	.	.	.	.
Gondola	10	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Goze	11	1	2	1	.	.	.	.	.	.	.	2	2	.	.	.	.	.	.	.	.	2	1	.
Gradi	12	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	1	.	.	.	.	3	.	.
Menze	13	.	.	.	.	.	1	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	1
Natali	14	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Pauli	15	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Poza	16	.	.	2	.	.	.	.	.	.	.	1	.	.	.	.	1	.	.	.	.	1	.	.
Ragnina	17	.	1	1	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	1	.	.
Resti	18	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	1
Saraca	19	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Slatarich	20	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Sorgo	21	.	2	1	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	1	1	.	1
Tudisi	22	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.
Zamagna	23	.	1	.	.	.	.	.	.	2	.	.	.	.	.	.	.	.	.	.	.	1	.	.

A Symmetric-Acyclic Decomposition of the Ragusan Families Network



Demo with Pajek

```
Read Network Tina.net
Net/Transform/Arcs-->Edges/Bidirected Only/Max
Draw/Draw
Layout/Energy/Kamada-Kawai/Free
Operations/Blockmodeling/Restricted Options [On]
Operations/Blockmodeling/Random Start
    [4, Ranks.MDL], [Repetitions, 100], [Clusters, 4], [RUN]
    extend the dialog box to see the model
Draw/Draw-Partition
```

Blockmodeling in 2-mode networks

We already presented some ways of rearranging 2-mode network matrices at the beginning of this lecture.

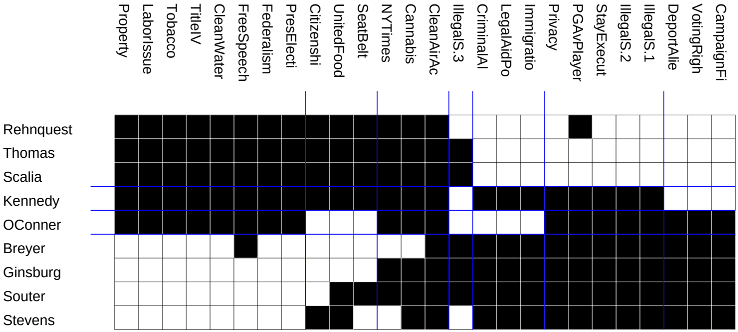
It is also possible to formulate this goal as a generalized blockmodeling problem where the solutions consist of two partitions — row-partition and column-partition.

Supreme Court Voting for Twenty-Six Important Decisions

Issue	Label	Br	Gi	So	St	OC	Ke	Re	Sc	Th
Presidential Election	PE	-	-	-	-	+	+	+	+	+
Criminal Law Cases										
Illegal Search 1	CL1	+	+	+	+	+	+	-	-	-
Illegal Search 2	CL2	+	+	+	+	+	+	-	-	-
Illegal Search 3	CL3	+	+	+	-	-	-	-	+	+
Seat Belts	CL4	-	-	+	-	-	+	+	+	+
Stay of Execution	CL5	+	+	+	+	+	+	-	-	-
Federal Authority Cases										
Federalism	FA1	-	-	-	-	+	+	+	+	+
Clean Air Action	FA2	+	+	+	+	+	+	+	+	+
Clean Water	FA3	-	-	-	-	+	+	+	+	+
Cannabis for Health	FA4	0	+	+	+	+	+	+	+	+
United Foods	FA5	-	-	+	+	-	+	+	+	+
NY Times Copyrights	FA6	-	+	+	-	+	+	+	+	+
Civil Rights Cases										
Voting Rights	CR1	+	+	+	+	+	-	-	-	-
Title VI Disabilities	CR2	-	-	-	-	+	+	+	+	+
PGA v. Handicapped Player	CR3	+	+	+	+	+	+	+	-	-
Immigration Law Cases										
Immigration Jurisdiction	Im1	+	+	+	+	-	+	-	-	-
Deporting Criminal Aliens	Im2	+	+	+	+	+	-	-	-	-
Detaining Criminal Aliens	Im3	+	+	+	+	-	+	-	-	-
Citizenship	Im4	-	-	-	+	-	+	+	+	+
Speech and Press Cases										
Legal Aid for Poor	SP1	+	+	+	+	-	+	-	-	-
Privacy	SP2	+	+	+	+	+	+	-	-	-
Free Speech	SP3	+	-	-	-	+	+	+	+	+
Campaign Finance	SP4	+	+	+	+	+	-	-	-	-
Tobacco Ads	SP5	-	-	-	-	+	+	+	+	+
Labor and Property Rights Cases										
Labor Rights	LPR1	-	-	-	-	+	+	+	+	+
Property Rights	LPR2	-	-	-	-	+	+	+	+	+

The Supreme Court Justices and their 'votes' on a set of 26 "important decisions" made during the 2000-2001 term, Doreian and Fujimoto (2002). The Justices (in the order in which they joined the Supreme Court) are: Rehnquist (1972), Stevens (1975), O'Conner (1981), Scalia (1982), Kennedy (1988), Souter (1990), Ginsburg (1993) and Breyer (1994).

…Supreme Court Voting / a (4,7) partition



upper – conservative / lower – liberal

Signed graphs

A *signed graph* is an ordered pair (G, σ) where

- $G = (V, R)$ is a directed graph (without loops) with set of vertices V and set of arcs $R \subseteq V \times V$;
- $\sigma : R \rightarrow \{p, n\}$ is a *sign* function. The arcs with the sign p are *positive* and the arcs with the sign n are *negative*. We denote the set of all positive arcs by R^+ and the set of all negative arcs by R^-.

The case when the graph is undirected can be reduced to the case of directed graph by replacing each edge e by a pair of opposite arcs both signed with the sign of the edge e.

Balanced and clusterable signed graphs

The signed graphs were introduced in Harary, 1953 and later studied by several authors. Following Roberts (1976, p. 75–77) a signed graph (G, σ) is:

- *balanced* iff the set of vertices V can be partitioned into two subsets so that every positive arc joins vertices of the same subset and every negative arc joins vertices of different subsets.
- *clusterable* iff the set of V can be partitioned into subsets, called *clusters*, so that every positive arc joins vertices of the same subset and every negative arc joins vertices of different subsets.

...Properties

The (semi)walk on the signed graph is *positive* iff it contains an even number of negative arcs; otherwise it is *negative*.
The balanced and clusterable signed graphs are characterised by the following theorems:
THEOREM 1. A signed graph (G, σ) is balanced iff every closed semiwalk is positive.

THEOREM 2. A signed graph (G, σ) is clusterable iff G contains no closed semiwalk with exactly one negative arc.

Balance semiring

To construct a semiring corresponding to the *balance* problem we take the set A with four elements: 0 – no walk; n – all walks are negative; p – all walks are positive; a – at least one positive and at least one negative walk.

$+$	0	n	p	a
0	0	n	p	a
n	n	n	a	a
p	p	a	p	a
a	a	a	a	a

$\cdot$	0	n	p	a
0	0	0	0	0
n	0	p	n	a
p	0	n	p	a
a	0	a	a	a

x	$x^\star$
0	p
n	a
p	p
a	a

The balance semiring is idempotent closed semiring with zero 0 and unit p.

Cluster semirings

For construction of the *cluster* semiring corresponding to the *clusterability* problem we need the set A with five elements: 0 – no walk; n – at least one walk with exactly one negative arc, no walk with only positive arcs; p – at least one walk with only positive arcs, no walk with exactly one negative arc; a – at least one walk with only positive arcs, at least one walk with exactly one negative arc; q – each walk has at least two negative arcs.

$+$	0	n	p	a	q
0	0	n	p	a	q
n	n	n	a	a	n
p	p	a	p	a	p
a	a	a	a	a	a
q	q	n	p	a	q

$\cdot$	0	n	p	a	q
0	0	0	0	0	0
n	0	q	n	n	q
p	0	n	p	a	q
a	0	n	a	a	q
q	0	q	q	q	q

x	$x^\star$
0	p
n	a
p	p
a	a
q	p

The cluster semiring is idempotent closed semiring with zero 0 and unit p.

Consequences

Let us define the *symmetric closure* of the value matrix by
$\mathbf{D}^\bullet = (\mathbf{D} + \mathbf{D}^T)^\star$ We get:

THEOREM 1'. A signed graph (G, σ) is balanced iff the diagonal of its balance-closure matrix $\mathbf{D}_B^\bullet$ contains only elements with value p.

THEOREM 2'. A signed graph (G, σ) is clusterable iff the diagonal of its cluster-closure matrix $\mathbf{D}_C^\bullet$ contains only elements with value p.

The balance-closure matrix of balanced signed graph contains no element with value a, since in this case the corresponding diagonal elements should also have value a. Similary the cluster-closure matrix of clusterable signed graph contains no element with value a.
A block is a maximal set of vertices with equal lines in matrix $\mathbf{D}^\bullet$.

... Consequences

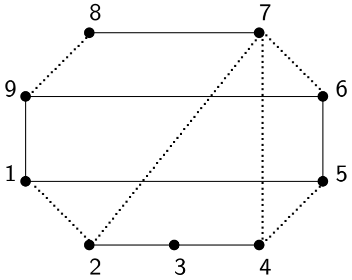
In balance-closure of balanced signed graph and in cluster-closure of clusterable signed graph all the entries between vertices of two blocks have the same value. The value of entries between vertices of the same block is p.
In both cases different partitions of the set of vertices correspond to the (nonequivalent) colorings of the graph with blocks as vertices in which there is an edge between two vertices iff the entries between the corresponding blocks in matrix $\mathbf{D}^\bullet$ have value n.

There is another way to test the clusterability of a given signed graph:

THEOREM 2". A signed graph (G, σ) is clusterable iff $(R^+)^\bullet \cap R^- = \emptyset$, where the closure $^\bullet$ is computed in the semiring $(\{0,1\}, \vee, \wedge, 0, 1)$.

This form of the Theorem 4 is interesting because the intersection $(R^+)^\bullet \cap R^-$ consists of arcs which prevent the signed graph (G, σ) to be clusterable.

Chartrand's example – graph



	1	2	3	4	5	6	7	8	9
1	0	*n*	0	0	*p*	0	0	0	*p*
2	*n*	0	*p*	0	0	0	*n*	0	0
3	0	*p*	0	*p*	0	0	0	0	0
4	0	0	*p*	0	*n*	0	*n*	0	0
5	*p*	0	0	*n*	0	*p*	0	0	0
6	0	0	0	0	*p*	0	*n*	0	*p*
7	0	*n*	0	*n*	0	*n*	0	*p*	0
8	0	0	0	0	0	0	*p*	0	*n*
9	*p*	0	0	0	0	*p*	0	*n*	0

In the figure the graph from Chartarand (1985, p. 181) and its value matrix are given. The positive edges are drawn with solid lines, and the negative edges with dotted lines.

Chartrand's example – closures

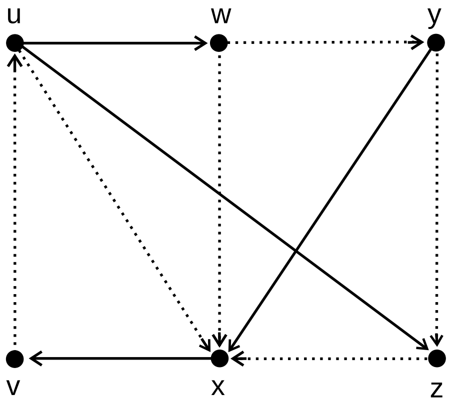
	1	2	3	4	5	6	7	8	9
1	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
2	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
3	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
4	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
5	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
6	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
7	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
8	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
9	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*

	1	2	3	4	5	6	7	8	9
1	**p**	*n*	*n*	*n*	**p**	**p**	*n*	*n*	**p**
2	*n*	**p**	**p**	**p**	*n*	*n*	*n*	*n*	*n*
3	*n*	**p**	**p**	**p**	*n*	*n*	*n*	*n*	*n*
4	*n*	**p**	**p**	**p**	*n*	*n*	*n*	*n*	*n*
5	**p**	*n*	*n*	*n*	**p**	**p**	*n*	*n*	**p**
6	**p**	*n*	*n*	*n*	**p**	**p**	*n*	*n*	**p**
7	*n*	*n*	*n*	*n*	*n*	*n*	**p**	**p**	*n*
8	*n*	*n*	*n*	*n*	*n*	*n*	**p**	**p**	*n*
9	**p**	*n*	*n*	*n*	**p**	**p**	*n*	*n*	**p**

On the left side of the table the corresponding balance-closure is given – the graph is not balanced. From the cluster-closure on the right side of the table we can see that the graph is clusterable and it has the clusters

$$V_1 = \{1, 5, 6, 9\}, \qquad V_2 = \{2, 3, 4\}, \qquad V_3 = \{7, 8\}$$

Roberts's example



In the figure the signed graph from Roberts (1976, page 77, exercise 16) is presented. Its value matrix on the left side of the following table.

Roberts's example – value matrix and its closure

	u	v	w	x	y	z
u	0	0	*p*	*n*	0	*p*
v	*n*	0	0	0	0	0
w	0	0	0	*n*	*n*	0
x	0	*p*	0	0	0	0
y	0	0	0	*p*	0	*n*
z	0	0	0	*n*	0	0

	u	v	w	x	y	z
u	**p**	*n*	**p**	*n*	*n*	**p**
v	*n*	**p**	*n*	**p**	**p**	*n*
w	**p**	*n*	**p**	*n*	*n*	**p**
x	*n*	**p**	*n*	**p**	**p**	*n*
y	*n*	**p**	*n*	**p**	**p**	*n*
z	**p**	*n*	**p**	*n*	*n*	**p**

In this case the balance-closure and the cluster-closure are equal (right side of the table). The corresponding partition is

$$V_1 = \{v, x, y\}, \qquad V_2 = \{u, w, z\}$$

Razcepnost označenih grafov in bločni modeli

Problemu razcepnosti ustrezajo tri vrste blokov:

- *ničelni* vsi elementi v bloku so enaki 0;
- *pozitivni* vsi elementi v bloku so pozitivni ali enaki 0;
- *negativni* vsi elementi v bloku so negativni ali enaki 0;

Če dana razvrstitev določa razcep omrežja, so diagonalni bloki pozitivni (ali ničelni), nediagonalni pa negativni ali ničelni.
Kakovost dane razvrstitve $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ torej lahko izmerimo takole ($0 \leq \alpha \leq 1$):

$$P_\alpha(\mathbf{C}) = \alpha \sum_{C\in\mathbf{C}} \sum_{u,v\in C} \max(0, -w_{uv}) + (1-\alpha) \sum_{\substack{C,C'\in\mathbf{C} \\ C\neq C'}} \sum_{u\in C, v\in C'} \max(0, w_{uv})$$

Za določitev čim boljše razvrstitve uporabimo lokalno optimizacijo.

Slovenske stranke 1994 (S. Kropivnik)

		1	2	3	4	5	6	7	8	9	10
SKD	1	0	-215	114	-89	-77	94	-170	176	117	-210
ZLSD	2	-215	0	-217	134	77	-150	57	-253	-230	49
SDSS	3	114	-217	0	-203	-80	138	-109	177	180	-174
LDS	4	-89	134	-203	0	157	-142	173	-241	-254	23
ZSESS	5	-77	77	-80	157	0	-188	170	-120	-160	-9
ZS	6	94	-150	138	-142	-188	0	-97	140	116	-106
DS	7	-170	57	-109	173	170	-97	0	-184	-191	-6
SLS	8	176	-253	177	-241	-120	140	-184	0	235	-132
SPS-SNS	9	117	-230	180	-254	-160	116	-191	235	0	-164
SNS	10	-210	49	-174	23	-9	-106	-6	-132	-164	0

Slovenske stranke 1994 / preurejene

		1	3	6	8	9	2	4	5	7	10
SKD	1	0	114	94	176	117	-215	-89	-77	-170	-210
SDSS	3	114	0	138	177	180	-217	-203	-80	-109	-174
ZS	6	94	138	0	140	116	-150	-142	-188	-97	-106
SLS	8	176	177	140	0	235	-253	-241	-120	-184	-132
SPS-SNS	9	117	180	116	235	0	-230	-254	-160	-191	-164
ZLSD	2	-215	-217	-150	-253	-230	0	134	77	57	49
LDS	4	-89	-203	-142	-241	-254	134	0	157	173	23
ZSESS	5	-77	-80	-188	-120	-160	77	157	0	170	-9
DS	7	-170	-109	-97	-184	-191	57	173	170	0	-6
SNS	10	-210	-174	-106	-132	-164	49	23	-9	-6	0

3-way blockmodeling

We started to work on blockmodeling of 3-way networks. We developed the indirect approach to *structural equivalence blockmodeling in 3-way networks*. *Indirect* means – embedding the notion of equivalence in a dissimilarity and determining it using clustering.

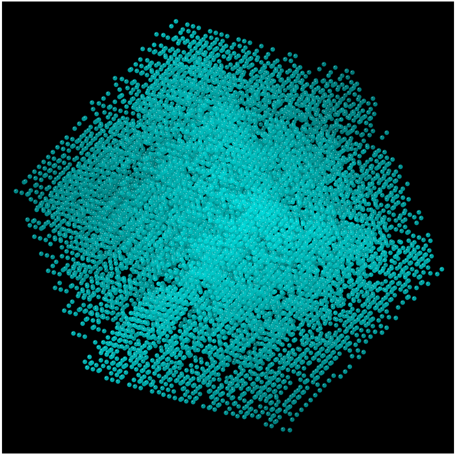
3-way network is defined by three sets of units X, Y and Z. There are three basic cases:

- all three sets are different (3-mode netork)
- two sets are the same (2-mode network)
- all three sets are the same (1-mode network)

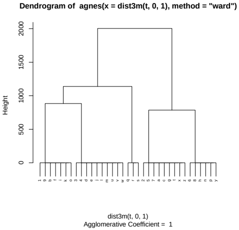
For all three cases we constructed compatible dissimilarities for structural equivalence .

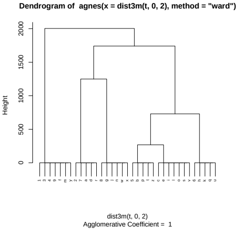
Example 1: Artificial dataset

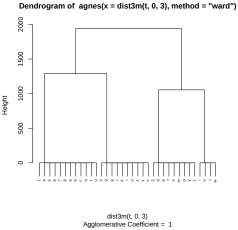
Randomly generated ideal structure `rndTest(c(5,6,4),c(35,35,35))`:



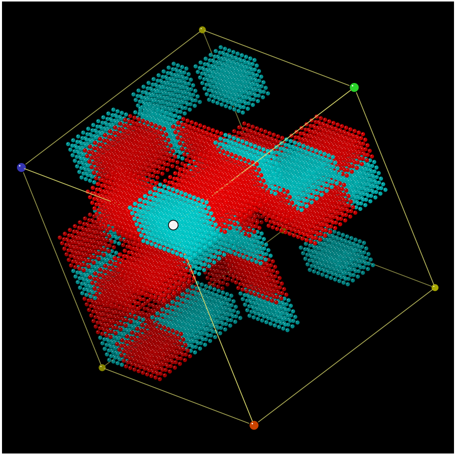
Example 1: Dendrograms





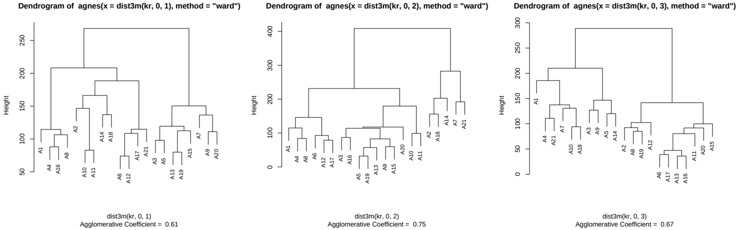


Example 1: Solutions

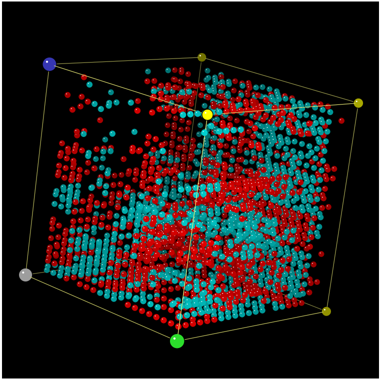


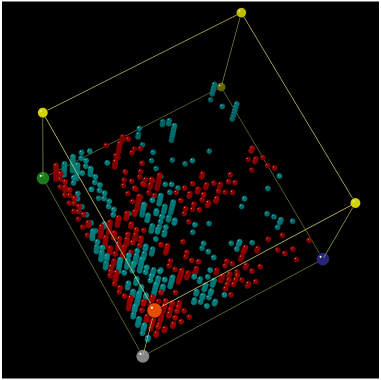
Click on the picture!

Example 2: Krackhardt / Dendrograms



Example 2: Krackhardt / Solutions

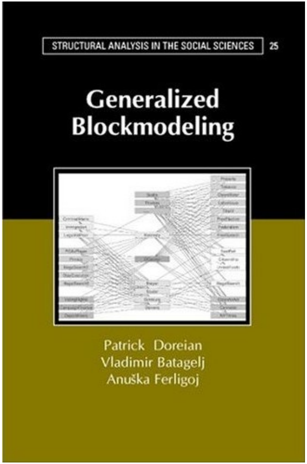




Blockmodeling of Valued Networks

Batagelj and Ferligoj (2000) proposed an approach to blockmodel of valued networks as an example of *relational data analysis*. These ideas were further developed by Žiberna (2007) who proposed some approaches for generalized blockmodeling of valued networks.
The first one is a straightforward generalization of the generalized blockmodeling of binary networks to valued blockmodeling. The second approach is homogeneity blockmodeling where the basic idea is that the inconsistency of an empirical block with its ideal block can be measured by within block variability of appropriate values. Žiberna provided new ideal blocks appropriate for blockmodeling of valued networks together with definitions of their block inconsistencies.

More on blockmodeling



The details about the generalized blockmodeling can be found in our book:
P. Doreian, V. Batagelj, A. Ferligoj:
Generalized Blockmodeling, CUP, 2005.

Conditions for hierarchical clustering methods

The set of feasible clusterings $\mathbf{\Phi}$ determines the *feasibility predicate* $\Phi(\mathbf{C}) \equiv \mathbf{C} \in \mathbf{\Phi}$ defined on $\mathcal{P}(\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\})$; and conversely $\mathbf{\Phi} \equiv \{\mathbf{C} \in \mathcal{P}(\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}) : \Phi(\mathbf{C})\}$.
In the set $\mathbf{\Phi}$ the relation of *clustering inclusion* $\sqsubseteq$ can be introduced by

$$\mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \equiv \forall C_1 \in \mathbf{C}_1, C_2 \in \mathbf{C}_2 : C_1 \cap C_2 \in \{\emptyset, C_1\}$$

we say also that the clustering $\mathbf{C}_1$ is a *refinement* of the clustering $\mathbf{C}_2$.
It is well known that $(\Pi(\mathcal{U}), \sqsubseteq)$ is a partially ordered set (even more, semimodular lattice). Because any subset of partially ordered set is also partially ordered, we have: Let $\mathbf{\Phi} \subseteq \Pi(\mathcal{U})$ then $(\mathbf{\Phi}, \sqsubseteq)$ is a partially ordered set.
The clustering inclusion determines two related relations (on $\mathbf{\Phi}$):
$\mathbf{C}_1 \sqsubset \mathbf{C}_2 \equiv \mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \wedge \mathbf{C}_1 \neq \mathbf{C}_2$ – strict inclusion, and
$\mathbf{C}_1 \sqsubset\!\!\!\cdot\, \mathbf{C}_2 \equiv \mathbf{C}_1 \sqsubset \mathbf{C}_2 \wedge \neg\exists \mathbf{C} \in \mathbf{\Phi} : (\mathbf{C}_1 \sqsubset \mathbf{C} \wedge \mathbf{C} \sqsubset \mathbf{C}_2)$ – predecessor.

Conditions on the structure of the set of feasible clusterings

We shall assume that the set of feasible clusterings $\mathbf{\Phi} \subseteq \Pi(\mathcal{U})$ satisfies the following conditions:

F1. $\mathbf{O} \equiv \{\{X\} : X \in \mathcal{U}\} \in \mathbf{\Phi}$

F2. The feasibility predicate Φ is *local* – it has the form $\Phi(\mathbf{C}) = \bigwedge_{C \in \mathbf{C}} \varphi(C)$ where $\varphi(C)$ is a predicate defined on $\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$ (clusters).
The intuitive meaning of $\varphi(C)$ is: $\varphi(C) \equiv$ the cluster C is 'good'.
Therefore the locality condition can be read: a 'good' clustering $\mathbf{C} \in \mathbf{\Phi}$ consists of 'good' clusters.

F3. The predicate Φ has the property of *binary heredity* with respect to the *fusibility* predicate $\psi(C_1, C_2)$, i.e.,

$$C_1 \cap C_2 = \emptyset \wedge \varphi(C_1) \wedge \varphi(C_2) \wedge \psi(C_1, C_2) \Rightarrow \varphi(C_1 \cup C_2)$$

This condition means: in a 'good' clustering, a fusion of two 'fusible' clusters produces a 'good' clustering.

. . . conditions

F4. The predicate ψ is *compatible* with clustering inclusion $\sqsubseteq$, i.e.,

$$\forall \mathbf{C}_1, \mathbf{C}_2 \in \mathbf{\Phi} : (\mathbf{C}_1 \sqsubset \mathbf{C}_2 \wedge \mathbf{C}_1 \setminus \mathbf{C}_2 = \{C_1, C_2\} \Rightarrow \psi(C_1, C_2) \vee \psi(C_2, C_1))$$

F5. The *interpolation* property holds in $\mathbf{\Phi}$, i.e., $\forall \mathbf{C}_1, \mathbf{C}_2 \in \mathbf{\Phi}$:

$$(\mathbf{C}_1 \sqsubset \mathbf{C}_2 \wedge \mathrm{card}(()\mathbf{C}_1) > \mathrm{card}(()\mathbf{C}_2) + 1 \Rightarrow \exists \mathbf{C} \in \mathbf{\Phi} : (\mathbf{C}_1 \sqsubset \mathbf{C} \wedge \mathbf{C} \sqsubset \mathbf{C}_2))$$

These conditions provide a framework in which the hierarchical methods can be applied also for constrained clustering problems $\mathbf{\Phi}_k(\mathcal{U}) \subset \Pi_k(\mathcal{U})$. In the ordinary problem both predicates $\varphi(C)$ and $\psi(C_p, C_q)$ are always true – all conditions F1-F5 are satisfied.

Clustering with relational constraint

Suppose that the units are described by attribute data $a \colon \mathcal{U} \to [\mathcal{U}]$ and related by a binary *relation* $R \subseteq \mathcal{U} \times \mathcal{U}$ that determine the *relational data* $(\mathcal{U}, R, a)$.
We want to cluster the units according to the similarity of their descriptions, but also considering the relation R – it imposes *constraints* on the set of feasible clusterings, usually in the following form:

$$\mathbf{\Phi}(R) = \{\mathbf{C} \in P(\mathcal{U}) : \text{each cluster } C \in \mathbf{C} \text{ is a subgraph } (C, R \cap C \times C) \\ \text{in the graph } (\mathcal{U}, R) \text{ of the required type of connectedness}\}$$

Example: regionalization problem – group given territorial units into regions such that units inside the region will be similar and form contiguous part of the territory.

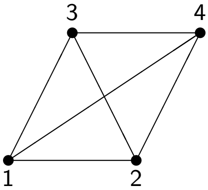
Some types of relational constraints

We can define different types of sets of feasible clusterings for the same relation R. Some examples of *types of relational constraint* $\Phi^i(R)$ are

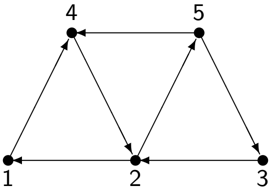
type of clusterings	type of connectedness
$\Phi^1(R)$	weakly connected units
$\Phi^2(R)$	weakly connected units that contain at most one center
$\Phi^3(R)$	strongly connected units
$\Phi^4(R)$	clique
$\Phi^5(R)$	the existence of a trail containing all the units of the cluster

Trail – all arcs are distinct.
A set of units $L \subseteq C$ is a *center* of cluster C in the clustering of type $\Phi^2(R)$ iff the subgraph induced by L is strongly connected and $R(L) \cap (C \setminus L) = \emptyset$.

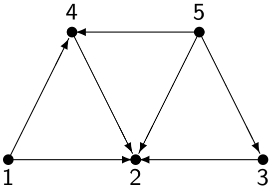
Some graphs of different types



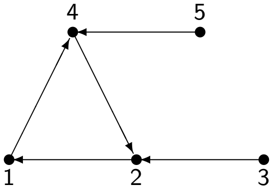
a clique



strongly connected units



weakly connected units



weakly connected units with a center $\{1, 2, 4\}$

Properties of relational constraints

The sets of feasible clusterings $\Phi^i(R)$ are linked as follows:

$\Phi^4(R) \subseteq \Phi^3(R) \subseteq \Phi^2(R) \subseteq \Phi^1(R)$

$\Phi^4(R) \subseteq \Phi^5(R) \subseteq \Phi^2(R)$

If the relation R is symmetric, then $\Phi^3(R) = \Phi^1(R)$

If the relation R is an equivalence relation, then $\Phi^4(R) = \Phi^1(R)$

Here are also examples of the corresponding fusibility predicates:

$\psi^1(C_1, C_2) \equiv \exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in C_2 : (\mathrm{X}R\mathrm{Y} \vee \mathrm{Y}R\mathrm{X})$

$\psi^2(C_1, C_2) \equiv (\exists \mathrm{X} \in L_1 \exists \mathrm{Y} \in C_2 : \mathrm{X}R\mathrm{Y}) \vee (\exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in L_2 : \mathrm{Y}R\mathrm{X})$

$\psi^3(C_1, C_2) \equiv (\exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in C_2 : \mathrm{X}R\mathrm{Y}) \wedge (\exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in C_2 : \mathrm{Y}R\mathrm{X})$

$\psi^4(C_1, C_2) \equiv \forall \mathrm{X} \in C_1 \forall \mathrm{Y} \in C_2 : (\mathrm{X}R\mathrm{Y} \wedge \mathrm{Y}R\mathrm{X})$

$\psi^5(C_1, C_2) \equiv (\exists \mathrm{X} \in T_1 \exists \mathrm{Y} \in I_2 : \mathrm{X}R\mathrm{Y}) \vee (\exists \mathrm{X} \in I_1 \exists \mathrm{Y} \in T_2 : \mathrm{Y}R\mathrm{X})$

For ψ^3 the property F5 fails.

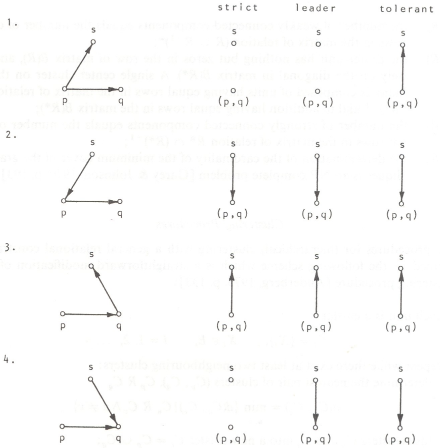
Agglomerative method for relational constraints

We can use both hierarchical and local optimization methods for solving some types of problems with relational constraint (Ferligoj, Batagelj 1983).

1. $k := n$; $\mathbf{C}(k) := \{\{X\} : X \in \mathcal{U}\}$;
2. **while** $\exists C_i, C_j \in \mathbf{C}(k)\colon (i \neq j \wedge \psi(C_i, C_j))$ **repeat**
   - 2.1. $(C_p, C_q) := \text{argmin}\{D(C_i, C_j)\colon i \neq j \wedge \psi(C_i, C_j)\}$;
   - 2.2. $C := C_p \cup C_q$; $k := k - 1$;
   - 2.3. $\mathbf{C}(k) := \mathbf{C}(k+1) \setminus \{C_p, C_q\} \cup \{C\}$;
   - 2.4. determine $D(C, C_s)$ for all $C_s \in \mathbf{C}(k)$
   - 2.4. adjust the relation R as required by the clustering type
3. $m := k$

The condition $\psi(C_i, C_j)$ is equivalent to $C_i R C_j$ for tolerant, leader and strict method; and to $C_i R C_j \wedge C_j R C_i$ for two-way method.

Adjusting relation after joining

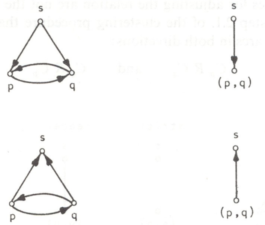


Φ^1 – tolerant

Φ^2 – leader

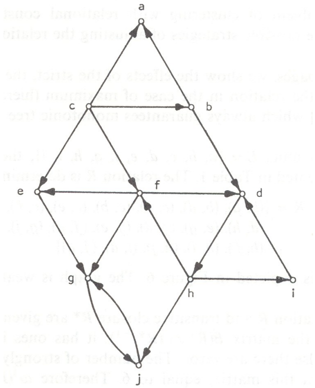
Φ^4 – two-way

Φ^5 – strict

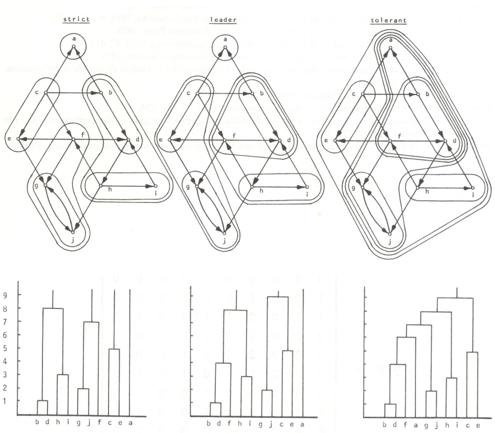


Example - problem

	a	b	c	d	e	f	g	h	i	j
a	0	5	7	4	6	6	2	4	2	3
b		0	8	1	3	4	4	5	3	4
c			0	4	5	7	9	3	2	5
d				0	3	2	4	8	6	3
e					0	6	4	6	5	7
f						0	6	8	5	7
g							0	4	8	2
h								0	3	4
i									0	5
j										0



Example - solution



Dissimilarities between clusters

In the original approach a complete dissimilarity matrix is needed. To obtain fast algorithms we propose to *consider only the dissimilarities between linked units*.

Let $(\mathcal{U}, R)$, $R \subseteq \mathcal{U} \times \mathcal{U}$ be a graph and $\emptyset \subset S, T \subset \mathcal{U}$ and $S \cap T = \emptyset$. We call a *block* of relation R for S and T its part $R(S, T) = R \cap S \times T$. The *symmetric closure* of relation R we denote with $\hat{R} = R \cup R^{-1}$. It holds: $\hat{R}(S, T) = \hat{R}(T, S)$.

For all dissimilarities between clusters $D(S, T)$ we set:

$$D(\{s\}, \{t\}) = \begin{cases} d(s, t) & s\hat{R}t \\ \infty & \text{otherwise} \end{cases}$$

where d is a selected dissimilarity between units.

Minimum and Maximum

Minimum

$$D_{\min}(S,T) = \min_{(s,t)\in\hat{R}(S,T)} d(s,t)$$

$$D_{\min}(S, T_1 \cup T_2) = \min(D_{\min}(S,T_1), D_{\min}(S,T_2))$$

Maximum

$$D_{\max}(S,T) = \max_{(s,t)\in\hat{R}(S,T)} d(s,t)$$

$$D_{\max}(S, T_1 \cup T_2) = \max(D_{\max}(S,T_1), D_{\max}(S,T_2))$$

Average

Average

$w : V \to \mathbb{R}$ – is a weight on units; for example $w(v) = 1$, for all $v \in \mathcal{U}$.

$$D_{\mathrm{a}}(S, T) = \frac{1}{w(\hat{R}(S, T))} \sum_{(s,t) \in \hat{R}(S,T)} d(s, t)$$

$$w(\hat{R}(S, T_1 \cup T_2)) = w(\hat{R}(S, T_1)) + w(\hat{R}(S, T_2))$$

$$D_{\mathrm{a}}(S, T_1 \cup T_2) = \frac{w(\hat{R}(S, T_1))}{w(\hat{R}(S, T_1 \cup T_2))} D_{\mathrm{a}}(S, T_1) + \frac{w(\hat{R}(S, T_2))}{w(\hat{R}(S, T_1 \cup T_2))} D_{\mathrm{a}}(S, T_2)$$

Hierarchies

The agglomerative clustering procedure produces a series of feasible clusterings $\mathbf{C}(n)$, $\mathbf{C}(n-1)$, ..., $\mathbf{C}(m)$ with $\mathbf{C}(m) \in \text{Max}\,\mathbf{\Phi}$ (maximal elements for $\sqsubseteq$).

Their union $\mathcal{T} = \bigcup_{k=m}^{n} \mathbf{C}(k)$ is called a *hierarchy* and has the property

$$\forall C_p, C_q \in \mathcal{T} : C_p \cap C_q \in \{\emptyset, C_p, C_q\}$$

The set inclusion $\subseteq$ is a *tree* or *hierarchical* order on $\mathcal{T}$. The hierarchy $\mathcal{T}$ is *complete* iff $\mathcal{U} \in \mathcal{T}$.

For $W \subseteq \mathcal{U}$ we define the *smallest cluster* $C_\mathcal{T}(W)$ from $\mathcal{T}$ containing W as:

c1. $W \subseteq C_\mathcal{T}(W)$

c2. $\forall C \in \mathcal{T} : (W \subseteq C \Rightarrow C_\mathcal{T}(W) \subseteq C)$

$C_\mathcal{T}$ is a *closure* on $\mathcal{T}$ with a special property

$$\text{Z} \notin C_\mathcal{T}(\{\text{X}, \text{Y}\}) \Rightarrow C_\mathcal{T}(\{\text{X}, \text{Y}\}) \subset C_\mathcal{T}(\{\text{X}, \text{Y}, \text{Z}\}) = C_\mathcal{T}(\{\text{X}, \text{Z}\}) = C_\mathcal{T}(\{\text{Y}, \text{Z}\})$$

Level functions

A mapping $h : \mathcal{T} \to \mathbb{R}_0^+$ is a *level function* on $\mathcal{T}$ iff
l1. $\forall \mathrm{X} \in \mathcal{U} : h(\{\mathrm{X}\}) = 0$
l2. $C_p \subseteq C_q \Rightarrow h(C_p) \leq h(C_q)$
A simple example of level function is $h(C) = \mathrm{card}(()C) - 1$.
Every hierarchy / level function determines an ultrametric dissimilarity on $\mathcal{U}$

$$\delta(\mathrm{X}, \mathrm{Y}) = h(C_{\mathcal{T}}(\{\mathrm{X}, \mathrm{Y}\}))$$

The converse is also true (see Dieudonne (1960)): Let d be an ultrametric on $\mathcal{U}$. Denote $\overline{B}(\mathrm{X}, r) = \{\mathrm{Y} \in \mathcal{U} : d(\mathrm{X}, \mathrm{Y}) \leq r\}$. Then for any given set $A \subset \mathbb{R}^+$ the set

$$\mathbf{C}(A) = \{\overline{B}(\mathrm{X}, r) : \mathrm{X} \in \mathcal{U}, r \in A\} \cup \{\{\mathcal{U}\}\} \cup \{\{\mathrm{X}\} : \mathrm{X} \in \mathcal{U}\}$$

is a complete hierarchy, and $h(C) = \mathrm{diam}(C)$ is a level function.

The pair $(\mathcal{T}, h)$ is called a *dendrogram* or a *clustering tree* because it can be visualized as a tree.

Reducibility

The dissimilarity D has the *reducibility* property (Bruynooghe, 1977) iff

$$D(C_p, C_q) \leq \min(D(C_p, C_s), D(C_q, C_s)) \;\Rightarrow$$

$$\min(D(C_p, C_s), d(C_q, C_s)) \leq D(C_p \cup C_q, C_s)$$

or equivalently

$$D(C_p, C_q) \leq t,\; D(C_p, C_s) \geq t,\; D(C_q, C_s) \geq t \;\Rightarrow\; D(C_p \cup C_q, C_s) \geq t$$

Theorem

If a dissimilarity D has the reducibility property then h_D is a level function.

Nearest neighbors graphs

For a given dissimilarity d on the set of units $\mathcal{U}$ and relational constraint R we define the *k nearest neighbors graph* $\mathbf{G}_{NN} = (\mathcal{U}, A)$

$$(\mathrm{X}, \mathrm{Y}) \in A \Leftrightarrow \mathrm{Y} \text{ is selected among the nearest neighbors of X and } \mathrm{X}\hat{R}\mathrm{Y}$$

By setting for $(\mathrm{X}, \mathrm{Y}) \in A$ its value to $w((\mathrm{X}, \mathrm{Y})) = d(\mathrm{X}, \mathrm{Y})$ we obtain a network $\mathcal{N}_{NN} = (\mathcal{U}, A, w)$.

In the case of equidistant pairs of units we have to decide – or to include them all in the graph, or specify an additional selection rule. We shall denote by $\mathbf{G}^*_{NN}$ the graph with included all equidistant pairs, and by $\mathbf{G}_{NN}$ a graph where a single nearest neighbor is always selected.

Structure and properties of the nearest neighbor graphs

Let $\mathcal{N}_{NN} = (\mathcal{U}, A, w)$ be a nearest neighbor network. A pair of units $\mathrm{X}, \mathrm{Y} \in \mathcal{U}$ are *reciprocal nearest neighbors* or RNNs iff $(\mathrm{X}, \mathrm{Y}) \in A$ and $(\mathrm{Y}, \mathrm{X}) \in A$.
Suppose $\mathrm{card}(()\mathcal{U}) > 1$ and R has no isolated units. Then in $\mathcal{N}$

- every unit/vertex $\mathrm{X} \in \mathcal{U}$ has the $\mathrm{outdeg}(\mathrm{X}) \geq 1$ — there is no isolated unit;
- along every walk the values of w are not increasing.

using these two observations we can show that in $\mathcal{N}^*_{NN}$:

- all the values of w on a closed walk are the same and all its arcs are reciprocal — all arcs between units in a nontrivial (at least 2 units) strong component are reciprocal;
- every maximal (can not be extended) elementary (no arc is repeated) walk ends in a RNNs pair;
- there exists at least one RNNs pair – corresponding to $\min_{\mathrm{X}, \mathrm{Y} \in \mathcal{U}, \mathrm{X} \neq \mathrm{Y}} d(\mathrm{X}, \mathrm{Y})$.

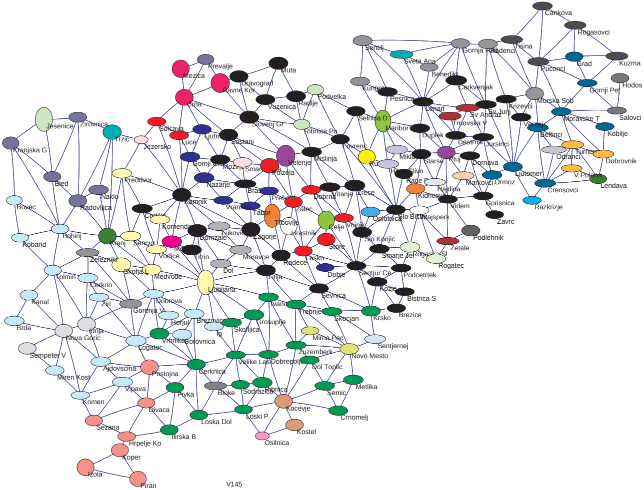
Fast agglomerative clustering algorithms

Any network $\mathcal{N}_{NN}$ is a subnetwork of $\mathcal{N}^*_{NN}$. Its connected components are directed (acyclic) trees with a single RNNs pair in the root.
Based on the nearest neighbor network very efficient algorithms for agglomerative clustering for methods with the reducibility property can be built.

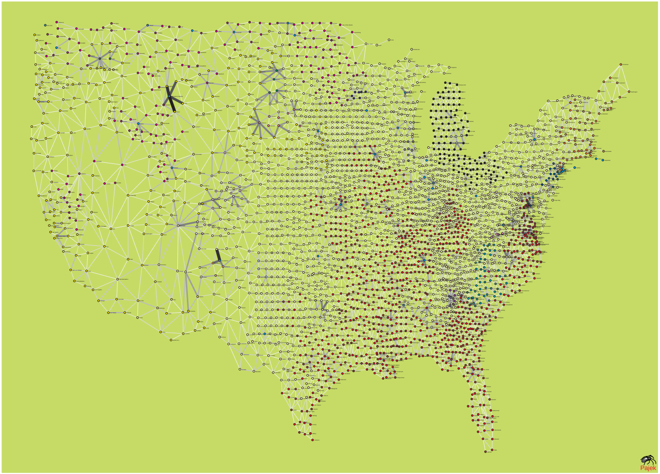
> *chain* := [ ]; **W** := $\mathcal{U}$;
> **while** card(()**W**) > 1 **do begin**
>   **if** *chain* = [ ] **then** select an arbitrary unit X ∈ **W** **else** X := *last*(*chain*);
>   grow a NN-chain from X until a pair (Y, Z) of RNNs are obtained;
>   agglomerate Y and Z:
> $T := \mathrm{Y} \cup \mathrm{Z}$; $\mathbf{W} := \mathbf{W} \setminus \{\mathrm{Y}, \mathrm{Z}\} \cup \{T\}$; compute $D(T, W), W \in \mathbf{W}$
> **end**;

It can be shown that if the clustering method has the reducibility property then the NN-chain remains a NN-chain also after the agglomeration of the RNNs pair.
`Network/Create Hierarchy/Clustering with Relational Constraint/`

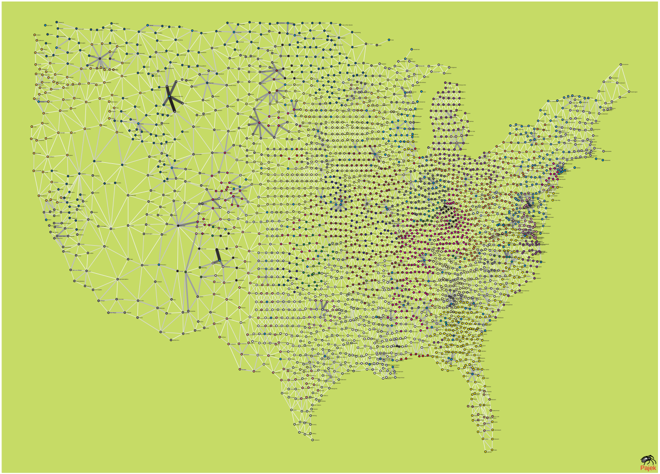
Example: Slovenian communes



Example: US counties $t = 1400$



Example: US counties $t = 200$



Louvain method and VOS

One of the approaches to determine the clustering **C** of a network is to maximize its *modularity*

$$Q(\mathbf{C}) = \sum_{C \in \mathbf{C}} \left(\frac{l(C)}{m} - \left(\frac{d(C)}{2m} \right)^2 \right)$$

where $l(C)$ is the number of edges between vertices belonging to cluster C, and $d(C)$ is the sum of the degrees of vertices from C.

The modularity maximization problem is NP complete.

Louvain method and VOS.

Final Remarks

The current, local optimization based, programs for generalized blockmodeling can deal only with networks with at most some hundreds of units. What to do with larger networks is an open question. For some specialized problems also procedures for (very) large networks can be developed (Doreian, Batagelj, Ferligoj, 1998; Batagelj, Zaveršnik, 2002).

Another interesting problem is the development of *blockmodeling of valued networks* or more general *relational data analysis* (Batagelj, Ferligoj, 2000).

References

- Batagelj V., Ferligoj A. (2000): Clustering relational data. Data Analysis (Eds.: W. Gaul, O. Opitz, M. Schader), Springer, Berlin, 3–15.
- Bruynooghe, M. (1977), Méthodes nouvelles en classification automatique des données taxinomiques nombreuses. Statistique et Analyse des Données, **3**, 24–42.
- Doreian, P., Batagelj, V., Ferligoj, A. (2000), *Symmetric-acyclic decompositions of networks*. *J. classif.*, **17**(1), 3–28.
- Ferligoj A., Batagelj V. (1982), Clustering with relational constraint. *Psychometrika*, **47**(4), 413–426.
- Ferligoj A., Batagelj V. (1983), Some types of clustering with relational constraints. *Psychometrika*, **48**(4), 541–552.
- Murtagh, F. (1985), Multidimensional Clustering Algorithms, *Compstat lectures,* **4**, Vienna: Physica-Verlag.