



Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Introduction to Network Analysis using **Pajek**

8. Clustering and Blockmodeling

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

PhD and MS program in Statistics
University of Ljubljana, 2022



Outline

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

- 1 Clustering**
- 2 Block modeling**
- 3 Generalized Blockmodeling**
- 4 Pre-specified blockmodeling**
- 5 Two-mode blockmodeling**
- 6 Signed graphs**
- 7 Generalizations**
- 8 Clustering with constraints**

TYPE OF MEMBERSHIP	Max. size	EVENTS AND PARTICIPATIONS													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
<i>Clique I:</i>	1	C	C	C	C	C	C	-	C	C					
Core.....	2	C	C	C	-	C	C	C	C	C	-				
	3	-	C	C	C	C	C	C	C	C	C	-			
	4	C	-	C	C	C	C	C	C	C	C	-			
	5			P	P	P	-	P	-	-	-				
Primary...	6			P	-	P	P	-	P	-					
	7				P	P	P	P	P	P	-				
Secondary.	8				-	S	-	S	S	S					
<i>Clique II:</i>	9				S	-	S	S	S	S					
Secondary.	10					S	S	S	S	-	-	S			
	11					-	P	P	P	-	P	P			
	12					-	P	P	P	-	P	P	P		
	13					C	C	C	C	-	C	C	C		
Core.....	14					C	-	C	C	C	C	C	C		
	15					C	C	-	C	C	C	C	C	C	
Secondary.	16					S	S	[S - S]	S	S					
	17					S	-	S	S	S					
	18					S	-	S	S	S					

Figure 2. Participation of the Southern Women in Events

Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si

Current version of slides (March 23, 2022 at 00:41): [slides PDF](#)



Clustering problem

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Let us start with the formal setting of the clustering problem. We shall use the following notation:

X – *unit*

X – *description* of unit X

\mathcal{U} – *space* of units

\mathcal{U} – finite *set of units*, $\mathcal{U} \subset \mathcal{U}$

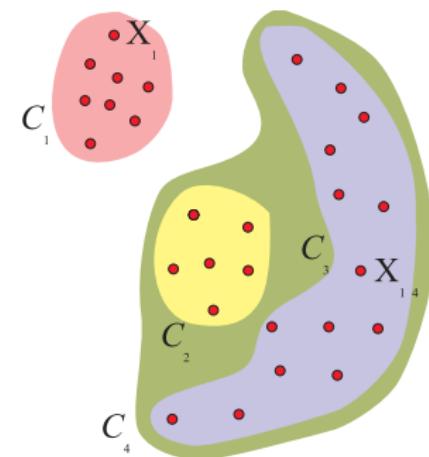
C – *cluster*, $\emptyset \subset C \subseteq \mathcal{U}$

\mathbf{C} – *clustering*, $\mathbf{C} = \{C_i\}$

Φ – set of *feasible clusterings*

P – *criterion function*,

$$P : \Phi \rightarrow \mathbb{R}_0^+$$





... Clustering problem

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

With these notions we can express the *clustering problem* (Φ, P) as follows:

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

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

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. (In theory) the set $\text{Min}(\Phi, P)$ can be determined by the complete search.

We shall denote the value of criterion function for an optimal clustering by $\min(\Phi, P)$.

Generally the clusters of clustering $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$ need not to be pairwise disjoint; yet, the clustering theory and practice mainly deal with clusterings which are the *partitions* of \mathcal{U} . We shall denote the set of all partitions of \mathcal{U} into k classes (clusters) by $\Pi_k(\mathcal{U})$.



Simple criterion functions

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Joining the individual units into a cluster C we make a certain "error", we create certain "tension" among them – we denote this quantity by $p(C)$. The *criterion function* $P(\mathbf{C})$ combines these "partial/local errors" into a "global error".

Usually it takes the form:

$$\text{S.} \quad P(\mathbf{C}) = \sum_{C \in \mathbf{C}} p(C) \quad \text{or} \quad \text{M.} \quad P(\mathbf{C}) = \max_{C \in \mathbf{C}} p(C)$$

For simple criterion functions usually
 $\min(\Pi_{k+1}\mathcal{U}, P) \leq \min(\Pi_k(\mathcal{U}), P)$
we fix the value of k and set $\Phi \subseteq \Pi_k(\mathcal{U})$.



Cluster-error function / dissimilarities

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The *cluster-error* $p(C)$ has usually the properties:

$$p(C) \geq 0 \quad \text{and} \quad \forall X \in \mathcal{U} : p(\{X\}) = 0$$

In the following we shall assume that these properties of $p(C)$ hold.

To express the cluster-error $p(C)$ we define on the space of units a *dissimilarity* $d : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}_0^+$ for which we require:

D1. $\forall X \in \mathcal{U} : d(X, X) = 0$

D2. *symmetric*: $\forall X, Y \in \mathcal{U} : d(X, Y) = d(Y, X)$

Usually the dissimilarity d is defined using another dissimilarity $\delta : [\mathcal{U}] \times [\mathcal{U}] \rightarrow \mathbb{R}_0^+$ as

$$d(X, Y) = \delta([X], [Y])$$



Cluster-error function / examples

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Now we can define several cluster-error functions:

$$S. \quad p(C) = \sum_{X, Y \in C, X < Y} w(X) \cdot w(Y) \cdot d(X, Y)$$

$$\bar{S}. \quad p(C) = \frac{1}{w(C)} \sum_{X, Y \in C, X < Y} w(X) \cdot w(Y) \cdot d(X, Y)$$

where $w : \mathcal{U} \rightarrow \mathbb{R}^+$ is a *weight* of units, which is extended to clusters by:

$$w(\{X\}) = w(X), \quad X \in \mathcal{U}$$

$$w(C_1 \cup C_2) = w(C_1) + w(C_2), \quad C_1 \cap C_2 = \emptyset$$

Often $w(X) = 1$ holds for each $X \in \mathcal{U}$. Then $w(C) = \text{card}(C)$.

$$M. \quad p(C) = \max_{X, Y \in C} d(X, Y) = \text{diam}(C) \quad - \text{diameter}$$

$$T. \quad p(C) = \min_{T \text{ is a spanning tree over } C} \sum_{(X:Y) \in T} d(X, Y)$$



Matrix rearrangement view on blockmodeling

Snyder & Kick's World trade network / $n = 118$, $m = 514$

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

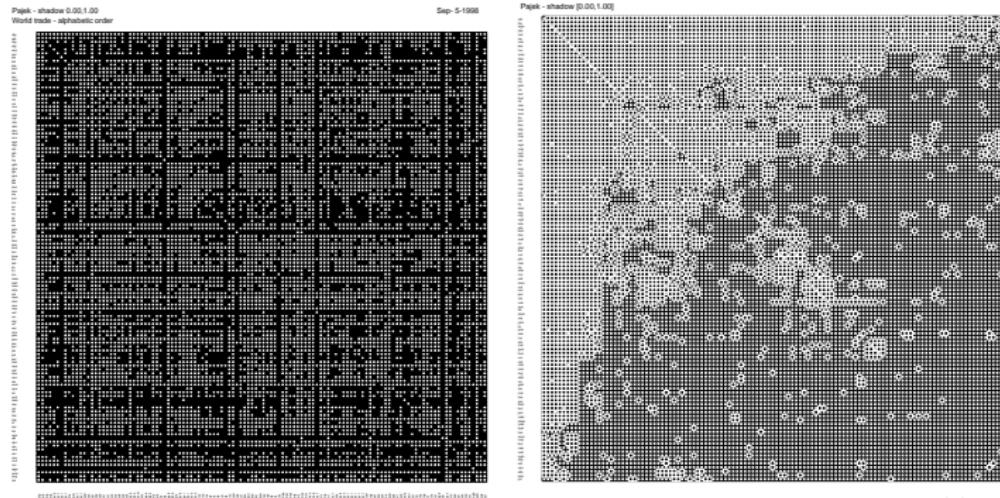
Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints



Alphabetic order of countries (left) and rearrangement (right)



Ordering the matrix

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

There are several ways how to rearrange a given matrix – determine an *ordering* or *permutation* of its rows and columns – to get some insight into its structure. For example:

- ordering by degree; by connected components; or by core number, connected components inside core levels, and degree;
- ordering according to a hierarchical clustering and some other property.

There exists also some special procedures to determine the ordering such as seriation and clumping (Murtagh) or **RCM** – Reverse Cuthill-McKee algorithm.

Partition/Make Permutation

Vector/Make Permutation

Network/Create Permutation/Reverse Cuthill-McKee

Network/Create Permutation/Core+Degree



Blockmodeling as a clustering problem

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

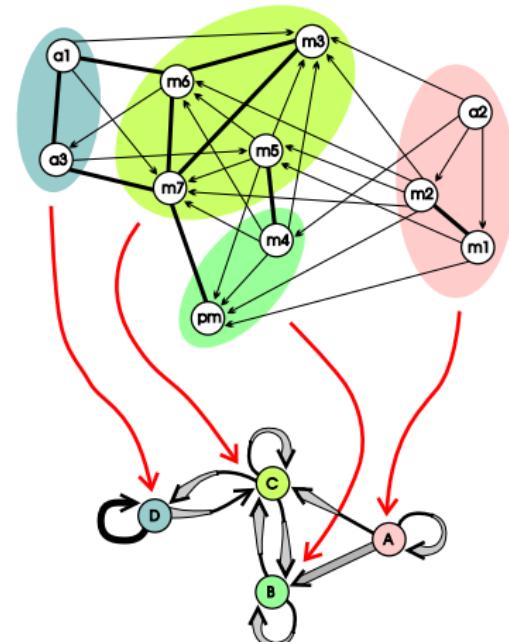
Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The goal of *blockmodeling* is to reduce a large, potentially incoherent network to a smaller comprehensible structure that can be interpreted more readily. Blockmodeling, as an empirical procedure, is based on the idea that units in a network can be grouped according to the extent to which they are equivalent, according to some *meaningful* definition of equivalence.





Cluster, clustering, blocks

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

One of the main procedural goals of blockmodeling is to identify, in a given network $\mathcal{N} = (\mathcal{U}, R)$, $R \subseteq \mathcal{U} \times \mathcal{U}$, **clusters** (classes) of units that share structural characteristics defined in terms of R .

The units within a cluster have the same or similar connection patterns to other units. They form a **clustering**

$\mathbf{C} = \{C_1, C_2, \dots, C_k\}$ which is a **partition** of the set \mathcal{U} . Each partition determines an equivalence relation (and vice versa). Let us denote by \sim the relation determined by partition **C**.

A clustering **C** partitions also the relation R into **blocks**

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

Each such block consists of units belonging to clusters C_i and C_j and all arcs leading from cluster C_i to cluster C_j . If $i = j$, a block $R(C_i, C_i)$ is called a **diagonal** block.



Structural and regular equivalence

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Regardless of the definition of equivalence used, there are two basic approaches to the equivalence of units in a given network (compare Faust, 1988):

- the equivalent units have the same connection pattern to the **same** neighbors;
- the equivalent units have the same or similar connection pattern to (possibly) **different** neighbors.

The first type of equivalence is formalized by the notion of structural equivalence and the second by the notion of regular equivalence with the latter a generalization of the former.



Structural equivalence

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Units are equivalent if they are linked to the rest of the network in *identical* ways (Lorrain and White, 1971). Such units are said to be *structurally equivalent*.

The units X and Y are *structurally equivalent*, we write $X \equiv Y$, iff the permutation (transposition) $\pi = (XY)$ is an automorphism of the relation R (Borgatti and Everett, 1992).

In other words, X and Y are structurally equivalent iff:

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



... structural equivalence

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0



Regular equivalence

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Integral to all attempts to generalize structural equivalence is the idea that units are equivalent if they link in equivalent ways to other units that are also equivalent.

White and Reitz (1983): The equivalence relation \approx on \mathcal{U} is a *regular equivalence* on network $\mathcal{N} = (\mathcal{U}, R)$ if and only if for all $X, Y, Z \in \mathcal{U}$, $X \approx Y$ implies both

$$R1. \quad XRZ \Rightarrow \exists W \in \mathcal{U} : (YRW \wedge W \approx Z)$$

$$R2. \quad ZRX \Rightarrow \exists W \in \mathcal{U} : (WRY \wedge W \approx Z)$$

Another view of regular equivalence is based on colorings (Everett, Borgatti 1996): regular equivalent nodes are of the same color and have the same set of colors in their neighborhoods.

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



... regular equivalence

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Theorem (Batagelj, Doreian, Ferligoj, 1992)

Let $\mathbf{C} = \{C_i\}$ be a partition corresponding to a regular equivalence \approx on the network $\mathcal{N} = (\mathcal{U}, R)$. Then each block $R(C_u, C_v)$ is either null or it has the property that there is at least one 1 in each of its rows and in each of its columns. Conversely, if for a given clustering \mathbf{C} , each block has this property then the corresponding equivalence relation is a regular equivalence.

The blocks for regular equivalence are null or 1-covered blocks.

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

1	0	1	0	0
0	0	1	0	1
0	1	0	0	0
1	0	1	1	0



Establishing Blockmodels

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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* (Φ, P):

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

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

where Φ is the set of *feasible clusterings* and P is a *criterion function*.

Criterion functions can be constructed

- *indirectly* as a function of a compatible (dis)similarity measure between pairs of units, or
- *directly* as a function measuring the fit of a clustering to an ideal one with perfect relations within each cluster and between clusters according to the considered types of connections (equivalence).



Indirect approach

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

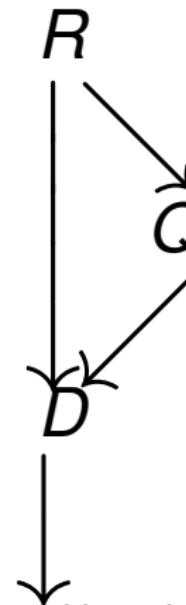
Clustering
with
constraints

RELATION

DESCRIPTIONS
OF UNITS

DISSIMILARITY
MATRIX

STANDARD
CLUSTERING
ALGORITHMS



original relation

path matrix
triads
orbits

hierarchical algorithms,
relocation algorithm, leader algorithm, etc.



Dissimilarities

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The property $t : \mathcal{U} \rightarrow \mathbb{R}$ 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 *triads* 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

$[u] = [t_1(u), t_2(u), \dots, t_m(u)]$. 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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

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



... 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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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
Network/Create Hierarchy/Clustering*/



... Dissimilarities

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

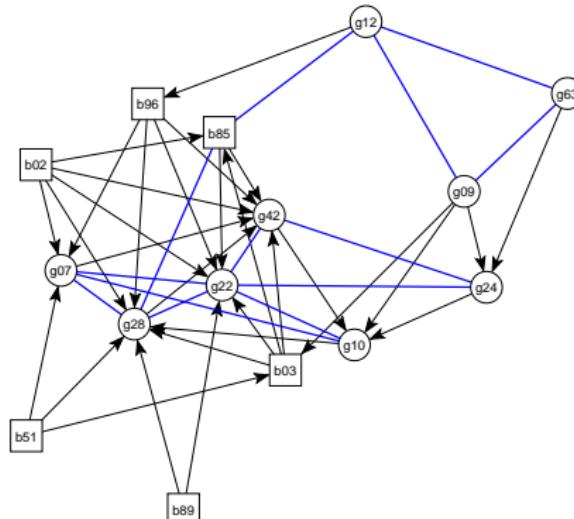
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



class.net

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

Cluster/Create Complete Cluster [n]

Operations/Network+Cluster/Dissimilarity*/Network Based/



Indirect approach

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

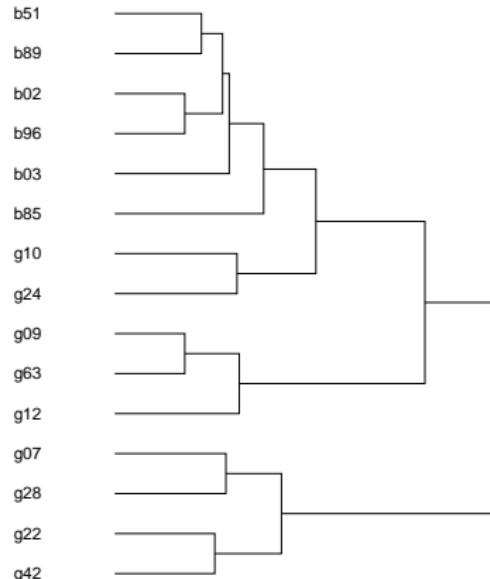
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

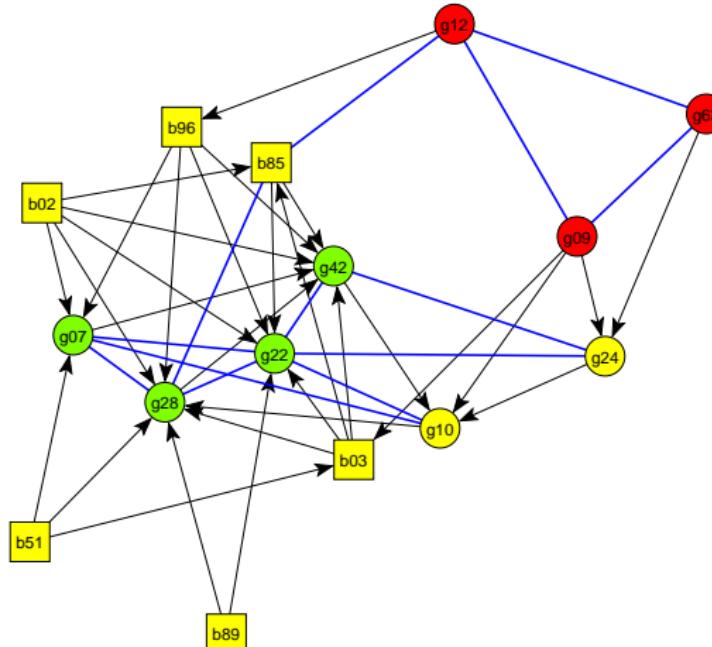
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



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



Matrix

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

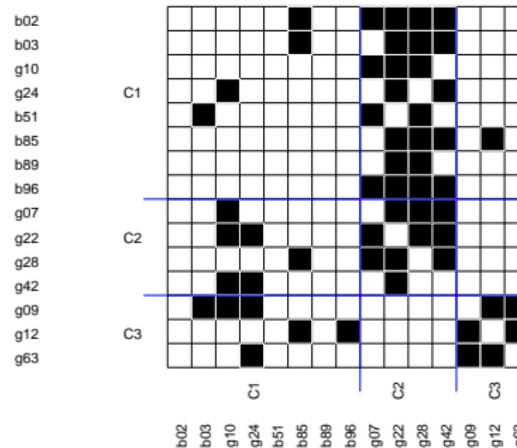
Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

Pajek - shadow [0.00,1.00]



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.

select cluster in the hierarchy

Hierarchy/Make permutation

Hierarchy/Make partition

File/Network/Export as Matrix to EPS/Using Permutation



Direct Approach and Generalized Blockmodeling

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

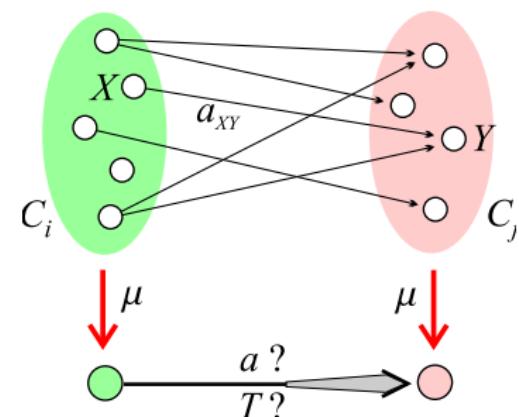
Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

A *blockmodel* consists of structures obtained by identifying all units from the same cluster of the clustering \mathbf{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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

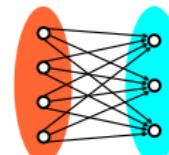
Two-mode
blockmodeling

Signed graphs

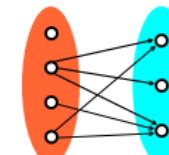
Generalizations

Clustering
with
constraints

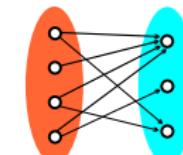
complete



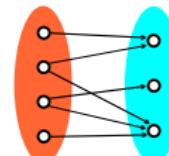
row-dominant



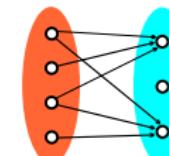
col-dominant



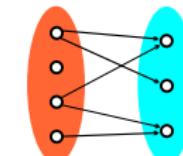
regular



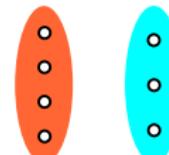
row-regular



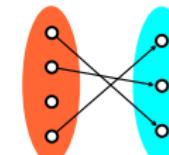
col-regular



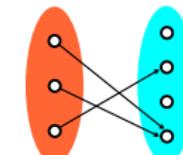
null



row-functional



col-functional





Generalized equivalence / Block Types

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

	Y		Y		Y
X	1 1 1 1 1	X	0 1 0 0 0	X	0 0 1 0 0
1 1 1 1 1		1 1 1 1 1	0 0 0 0 0	0 0 1 1 0	1 1 1 0 0
1 1 1 1 1		0 0 0 0 0	0 0 0 1 0	0 0 1 0 1	
complete	row-dominant	col-dominant			
	Y		Y		Y
X	0 1 0 0 0	X	0 1 0 0 0	X	0 1 0 1 0
1 0 1 1 0		0 1 1 0 0	1 0 1 0 0	1 0 1 0 0	1 1 0 1 1
0 0 1 0 1		1 0 1 0 0	0 1 0 0 1	0 0 0 0 0	
1 1 0 0 0		0 1 0 0 1			
regular	row-regular	col-regular			
	Y		Y		Y
X	0 0 0 0 0	X	0 0 0 1 0	X	1 0 0 0 0
0 0 0 0 0		0 0 1 0 0	1 0 0 0 0	0 1 0 0 0	0 0 1 0 0
0 0 0 0 0		1 0 0 0 0	0 0 0 1 0	0 0 0 0 0	0 0 0 0 1
0 0 0 0 0		0 0 0 1 0			
null	row-functional	col-functional			



Characterizations of Types of Blocks

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

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	\exists all 1 row *	
col-dominant	cdo	\exists all 1 column *	
row-functional	rfn	$\exists!$ one 1 in each row	
col-functional	cfn	$\exists!$ one 1 in each column	
non-null	one	\exists at least one 1	

* except this may be diagonal



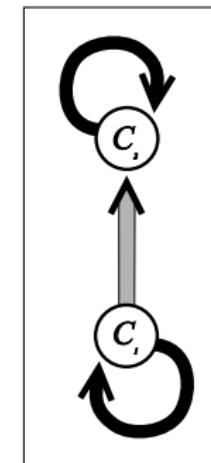
Block Types and Matrices

Clustering

V. Batagelj

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
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

	C_1	C_2
C_1	complete	regular
C_2	null	complete





Formalization of blockmodeling

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs
Generalizations

Clustering
with
constraints

Let V be a set of positions or images of clusters of units. Let $\mu : \mathcal{U} \rightarrow V$ 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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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 $\pi : K \rightarrow \mathcal{T} \setminus \{\text{nul}\}$ assigns predicates to connections;
- Q is a set of *averaging rules*. A mapping $\alpha : K \rightarrow Q$ determines rules for computing values of connections.

A (surjective) mapping $\mu : \mathcal{U} \rightarrow V$ determines a blockmodel \mathcal{M} of network \mathcal{N} iff it satisfies the conditions:

$$\forall (t, w) \in K : \pi(t, w)(C(t), C(w)) \text{ and}$$

$$\forall (t, w) \in V \times V \setminus K : \text{nul}(C(t), C(w)) .$$



Equivalences

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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 $\mathcal{T} = \{\text{nul, reg}\}$ reduces to the usual definition of regular equivalence (White and Reitz 1983). Similarly, compatibility for $\mathcal{T} = \{\text{nul, com}\}$ reduces to structural equivalence (Lorrain and White 1971).

For a compatible equivalence \sim the mapping $\mu: X \mapsto [X]$ 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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

One of 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 $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$, 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 functions

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

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

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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 = $\text{card}(X) - \# \text{ of rows in a block}$,
- n_c = $\text{card}(Y) - \# \text{ of columns in a block}$,
- 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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Connection	$\delta(X, Y; T)$	
null	$\begin{cases} s_t \\ s_t + d - s_d \end{cases}$	nondiagonal diagonal
complete	$\begin{cases} n_r n_c - s_t \\ n_r n_c - s_t + d + s_d - n_r \end{cases}$	nondiagonal diagonal
row-dominant	$\begin{cases} (n_c - m_r - 1)n_r \\ (n_c - m_r)n_r \end{cases}$	diagonal, $s_d = 0$ otherwise
col-dominant	$\begin{cases} (n_r - m_c - 1)n_c \\ (n_r - m_c)n_c \end{cases}$	diagonal, $s_d = 0$ 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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The obtained optimization problem can be solved by local optimization.

Network/Create Partition/Blockmodeling*/

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

- *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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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:

*	*
*	0

center -
periphery

*	0
*	*

hierarchy

*	0
0	*

clustering

0	*
*	0

bipartition



Prespecified blockmodeling example

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

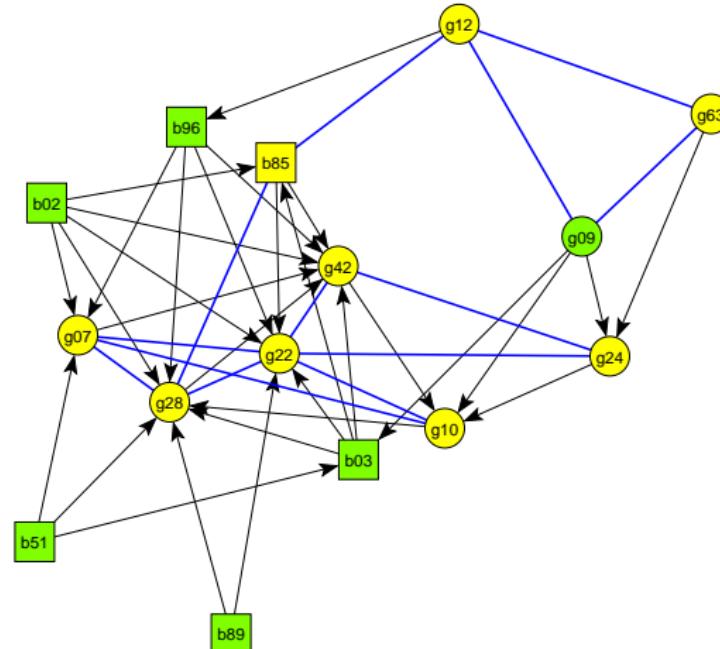
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Model

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

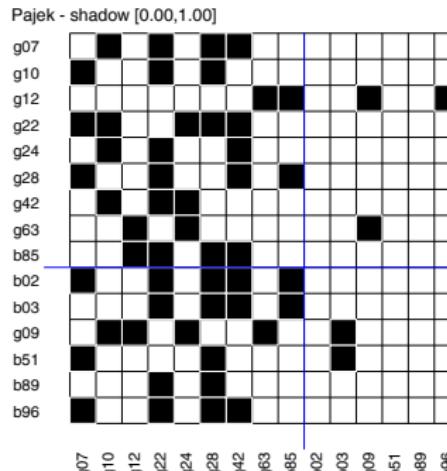


Image and Error Matrices:

	1	2		1	2
1	reg	-	1	0	3
2	reg	-	2	0	2

Total error = 5

Operations/Network+Partition/Blockmodeling*/Random Start



The Student Government at the University of Ljubljana in 1992

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

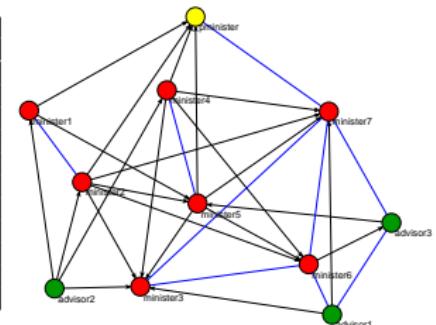
Generalizations

Clustering
with
constraints

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	.	.	.	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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

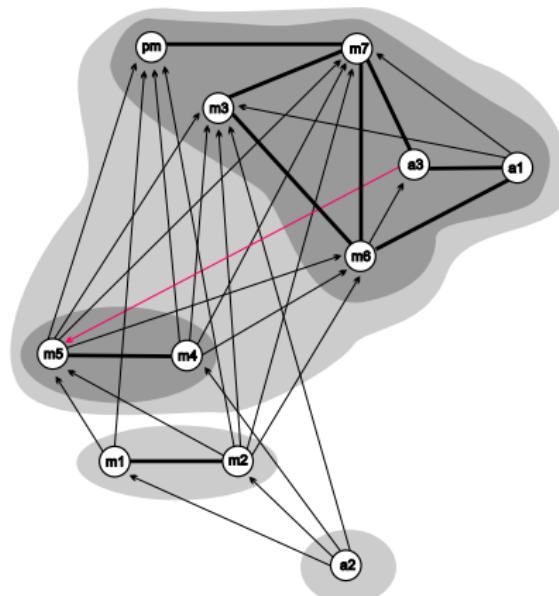
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



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

Network/Create Hierarchy/Symmetric-Acyclic



Football

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

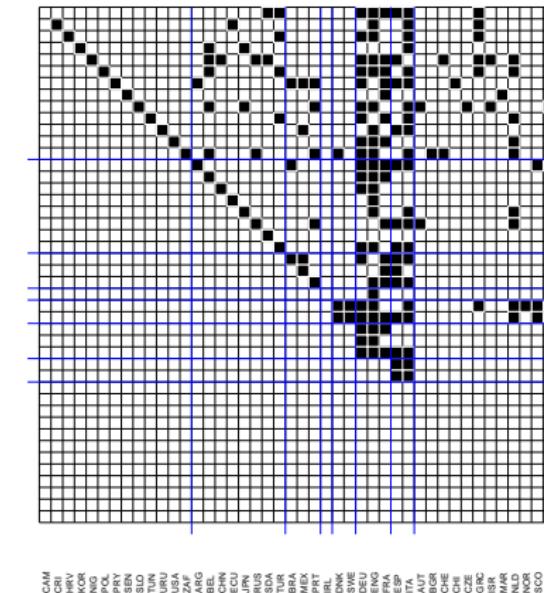
Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

Pajek - shadow [0.00,1.00]



The player's market of the Fifa Football Worldchampionship 2002 (Japan/Korea). The data, collected by L. Krempel, describe the 733 players of all 32 participating national teams and the clubs and countries where each of these players have contracts.

For acyclic (below diagonal blocks are zero-blocks) regular blockmodel we get a solution with 8 clusters and Error = 30



Demo with Pajek

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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



Blockmodeling in 2-mode networks

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

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'Connor (1981), Scalia (1982), Kennedy (1988), Souter (1990), Ginsburg (1993) and Breyer (1994).



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

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

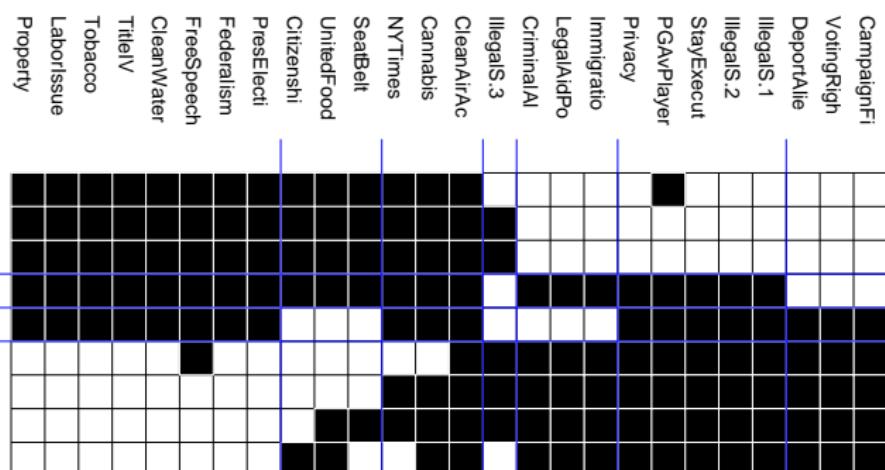
Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints



upper – conservative / lower – liberal



Signed graphs

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

A *signed graph* is an ordered pair (\mathcal{G}, σ) where

- $\mathcal{G} = (\mathcal{V}, R)$ is a directed graph (without loops) with set of nodes \mathcal{V} and set of arcs $R \subseteq \mathcal{V} \times \mathcal{V}$;
- $\sigma : R \rightarrow \{p, n\}$ 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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

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



... Properties

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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 (\mathcal{G}, σ) is balanced iff every closed semiwalk is positive.

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



Chartrand's example – graph

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

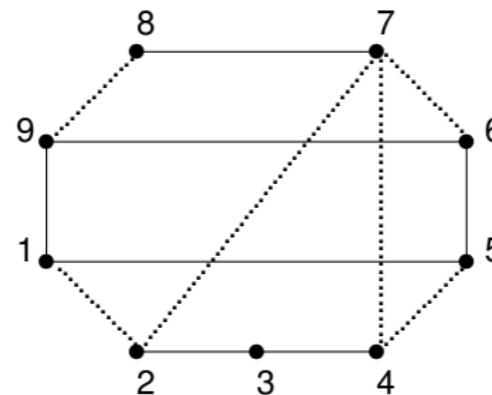
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



	1	2	3	4	5	6	7	8	9
1	0	<i>n</i>	0	0	<i>p</i>	0	0	0	<i>p</i>
2	<i>n</i>	0	<i>p</i>	0	0	0	<i>n</i>	0	0
3	0	<i>p</i>	0	<i>p</i>	0	0	0	0	0
4	0	0	<i>p</i>	0	<i>n</i>	0	<i>n</i>	0	0
5	<i>p</i>	0	0	<i>n</i>	0	<i>p</i>	0	0	0
6	0	0	0	0	<i>p</i>	0	<i>n</i>	0	<i>p</i>
7	0	<i>n</i>	0	<i>n</i>	0	<i>n</i>	0	<i>p</i>	0
8	0	0	0	0	0	0	<i>p</i>	0	<i>n</i>
9	<i>p</i>	0	0	0	0	<i>p</i>	0	<i>n</i>	0

In the figure the graph from Chartrand (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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

	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

$$\mathcal{V}_1 = \{1, 5, 6, 9\}, \quad \mathcal{V}_2 = \{2, 3, 4\}, \quad \mathcal{V}_3 = \{7, 8\}$$



Clusterability and blockmodeling

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

To the sign graph clusterability problem correspond three types of blocks:

- *null* all elements in a block are 0;
- *positive* all elements in a block are positive or 0;
- *negative* all elements in a block are negative or 0;

If a graph is clusterable the blocks determined by the partition are: positive or null on the diagonal; and negative or null outside the diagonal.

The clusterability of partition $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$ can be therefore measured as follows ($0 \leq \alpha \leq 1$):

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

The blockmodeling problem can be solved by local optimization.



Slovenian political parties 1994 (S. Kropivnik)

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

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

SKD – Slovene Christian Democrats; ZLSD – Associated List of Social Democrats;

SDSS – Social Democratic Party of Slovenia; LDS – Liberal Democratic Party;

ZSESS and ZS – two Green Parties, separated after 1992 elections; DS – Democratic Party;

SLS – Slovene People's Party; SNS – Slovene National Party;

SPS SNS – a group of deputies, former members of SNS, separated after 1992 elections

Network **Stranke94**.



Slovenian political parties 1994 / reordered

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

		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

S. Kropivnik, A. Mrvar: An Analysis of the Slovene Parliamentary Parties Network. in Developments in data analysis, MZ 12, FDV, Ljubljana, 1996, p. 209-216.



3-way blockmodeling

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

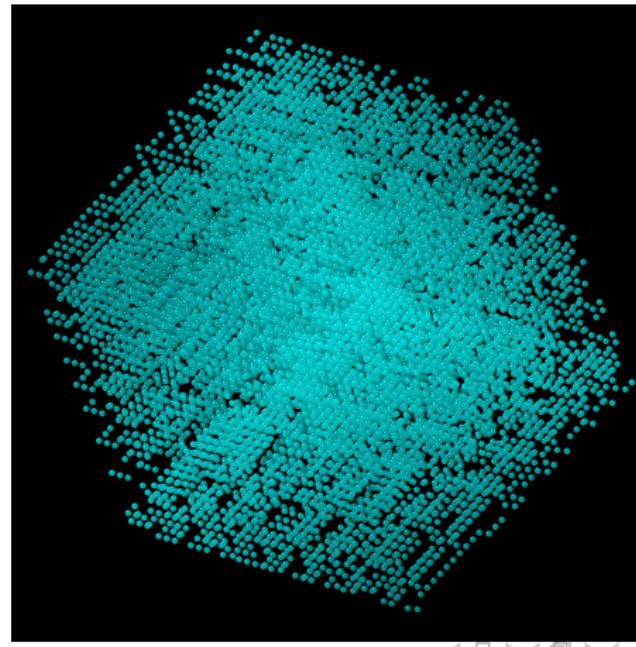
Signed graphs

Generalizations

Clustering
with
constraints

Randomly generated ideal structure

rndTest (c(5, 6, 4), c(35, 35, 35)):





Example 1: Dendrograms

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

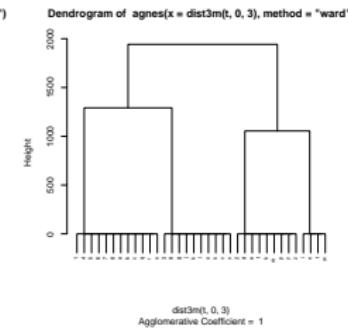
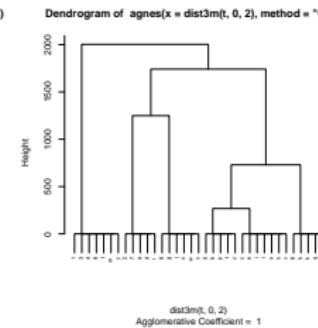
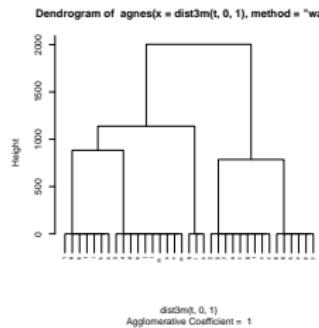
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Example 1: Solutions

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

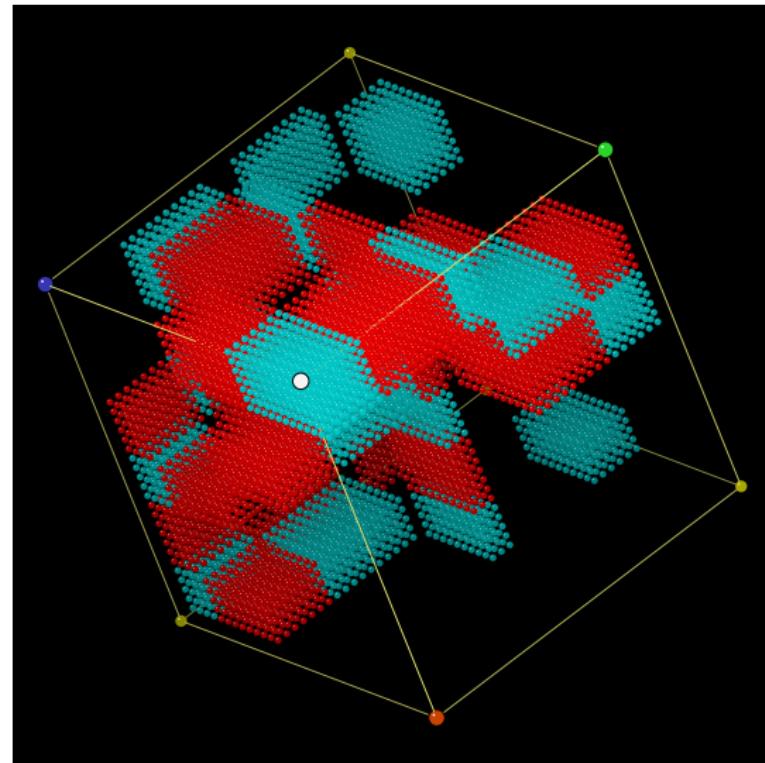
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Example 2: Krackhardt / Dendrograms

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

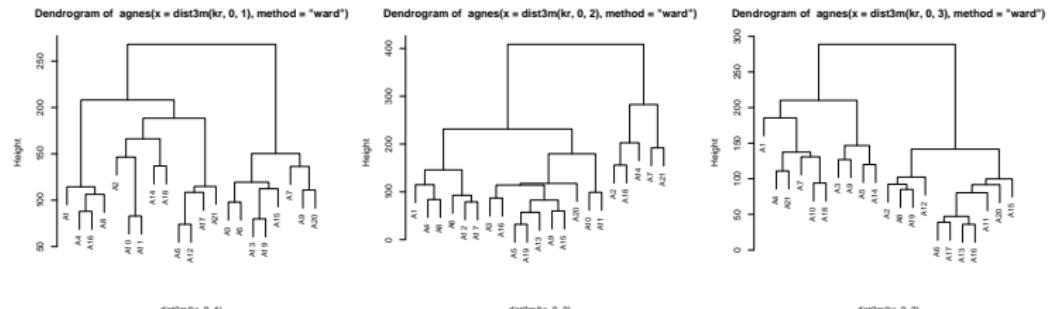
Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints





Example 2: Krackhardt / Solutions

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

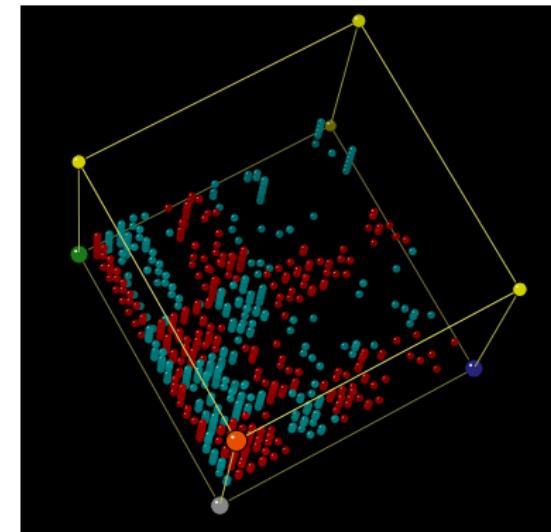
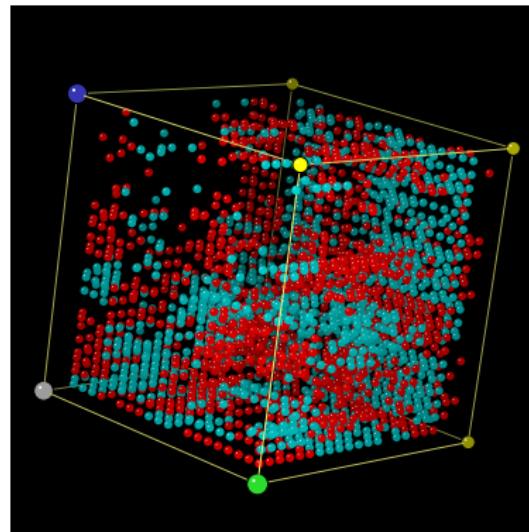
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Blockmodeling of Valued Networks

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

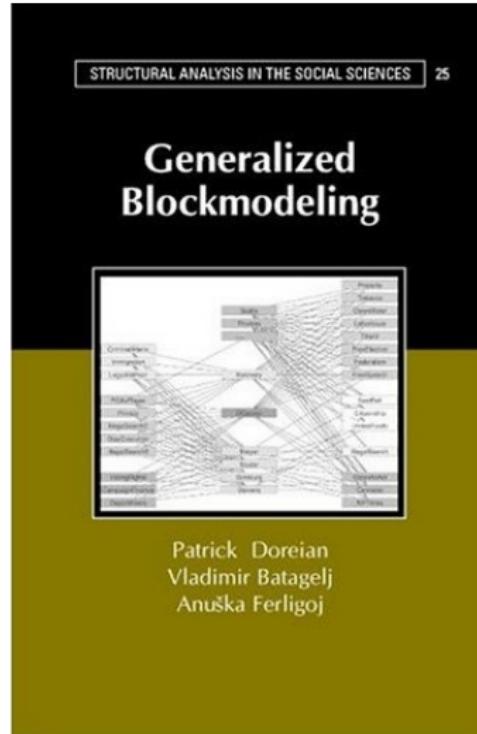
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints



The details about the generalized blockmodeling can be found in our book:

P. Doreian, V. Batagelj, A. Ferligoj: *Generalized Blockmod-
eling*, CUP, 2005.



Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

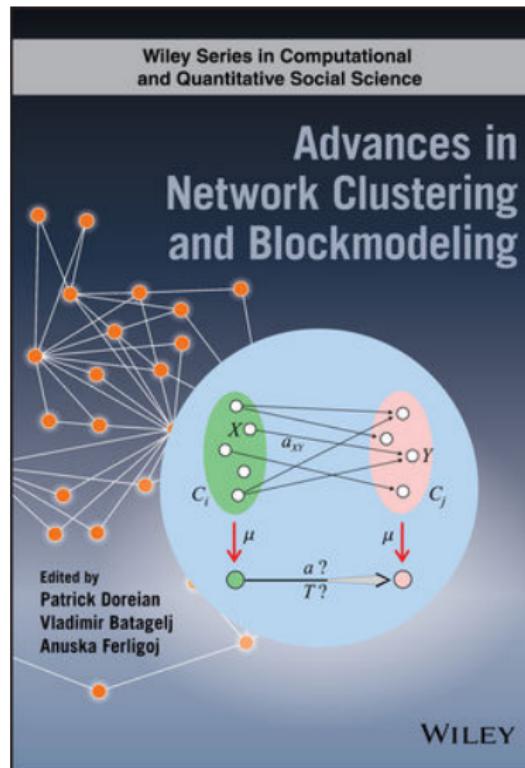
Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

... More on blockmodeling



Wiley 2020
Amazon



Final Remarks

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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).



Conditions for hierarchical clustering methods

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

The set of feasible clusterings Φ determines the *feasibility predicate* $\Phi(\mathbf{C}) \equiv \mathbf{C} \in \Phi$ defined on $\mathcal{P}(\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\})$; and conversely $\Phi \equiv \{\mathbf{C} \in \mathcal{P}(\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}) : \Phi(\mathbf{C})\}$.

In the set Φ 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 $\Phi \subseteq \Pi(\mathcal{U})$ then (Φ, \sqsubseteq) is a partially ordered set.

The clustering inclusion determines two related relations (on Φ):

$\mathbf{C}_1 \sqsubset \mathbf{C}_2 \equiv \mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \wedge \mathbf{C}_1 \neq \mathbf{C}_2$ – strict inclusion, and

$\mathbf{C}_1 \sqleftarrow \mathbf{C}_2 \equiv \mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \wedge \neg \exists \mathbf{C} \in \Phi : (\mathbf{C}_1 \sqsubset \mathbf{C} \wedge \mathbf{C} \sqsubset \mathbf{C}_2)$ – predecessor.



Conditions on the structure of the set of feasible clusterings

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

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

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

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 $\mathbf{C} \in \Phi$ 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.,

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

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

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

These conditions provide a framework in which the hierarchical methods can be applied also for constrained clustering problems $\Phi_k(\mathcal{U}) \subset \Pi_k(\mathcal{U})$.

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

Suppose that the units are described by attribute data $a: \mathcal{U} \rightarrow [\mathcal{U}]$ 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:

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

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
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	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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

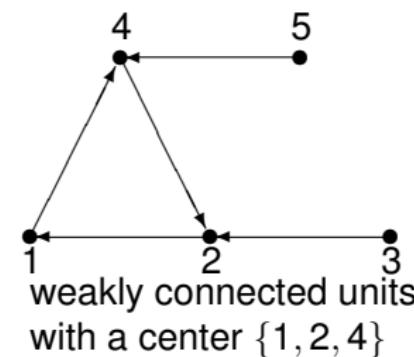
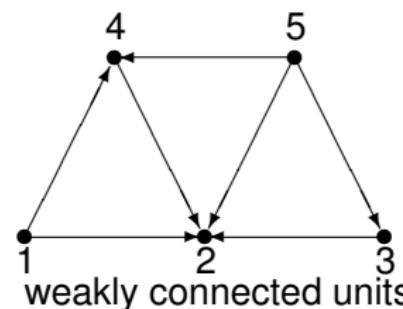
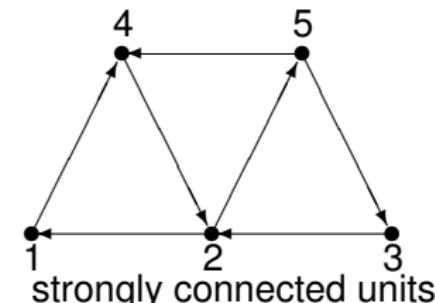
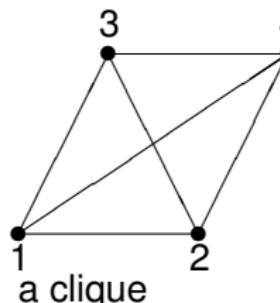
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Properties of relational constraints

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
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:

$$\psi^1(C_1, C_2) \equiv \exists X \in C_1 \exists Y \in C_2 : (XRY \vee YRX)$$

$$\psi^2(C_1, C_2) \equiv (\exists X \in L_1 \exists Y \in C_2 : XRY) \vee (\exists X \in C_1 \exists Y \in L_2 : YRX)$$

$$\psi^3(C_1, C_2) \equiv (\exists X \in C_1 \exists Y \in C_2 : XRY) \wedge (\exists X \in C_1 \exists Y \in C_2 : YRX)$$

$$\psi^4(C_1, C_2) \equiv \forall X \in C_1 \forall Y \in C_2 : (XRY \wedge YRX)$$

$$\psi^5(C_1, C_2) \equiv (\exists X \in T_1 \exists Y \in I_2 : XRY) \vee (\exists X \in I_1 \exists Y \in T_2 : YRX)$$

For ψ^3 the property F5 fails.



Agglomerative method for relational constraints

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
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** $\exists C_i, C_j \in \mathbf{C}(k) : (i \neq j \wedge \psi(C_i, C_j))$ **repeat**
 - 2.1. $(C_p, C_q) := \operatorname{argmin}\{D(C_i, C_j) : i \neq j \wedge \psi(C_i, C_j)\};$
 - 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 $C_i RC_j$ for tolerant, leader and strict method; and to $C_i RC_j \wedge C_j RC_i$ for two-way method.



Adjusting relation after joining

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

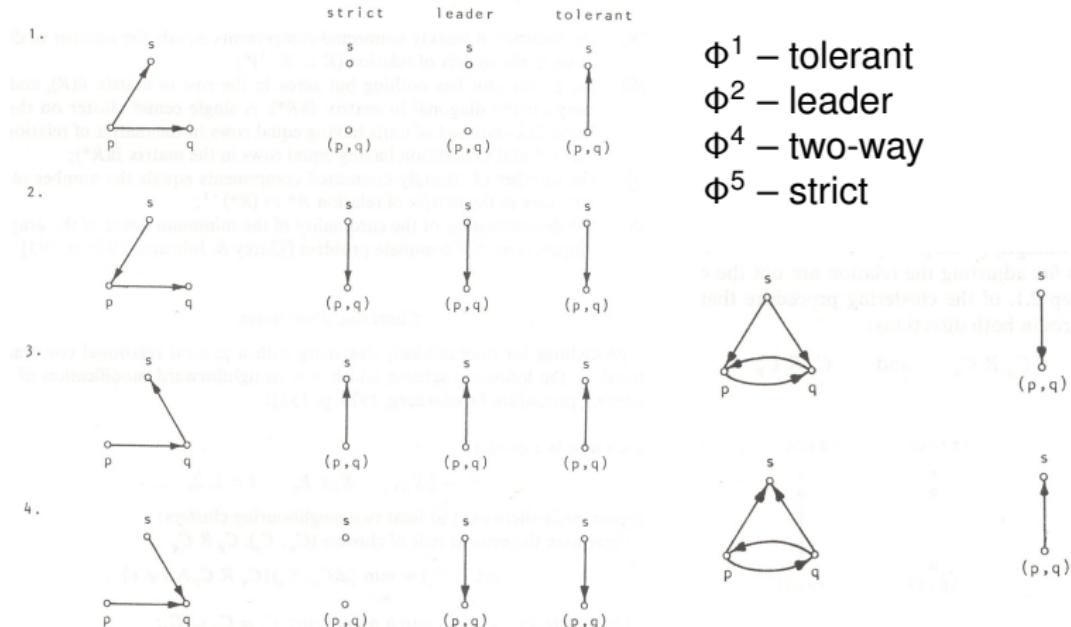
Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

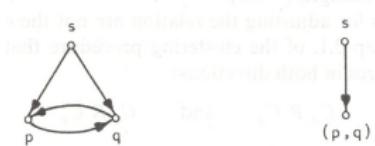


Φ^1 – tolerant

Φ^2 – leader

Φ^4 – two-way

Φ^5 – strict





Example - problem

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

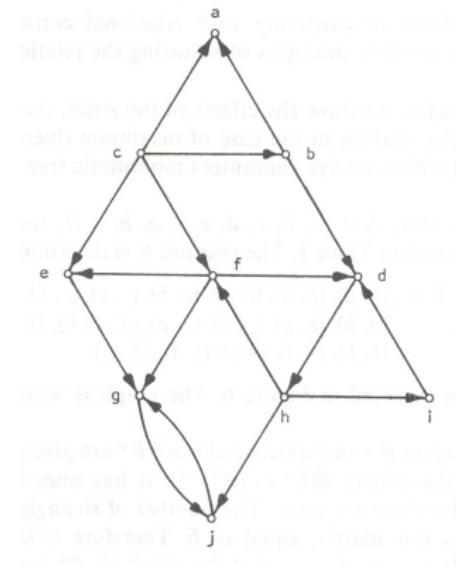
Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

	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

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

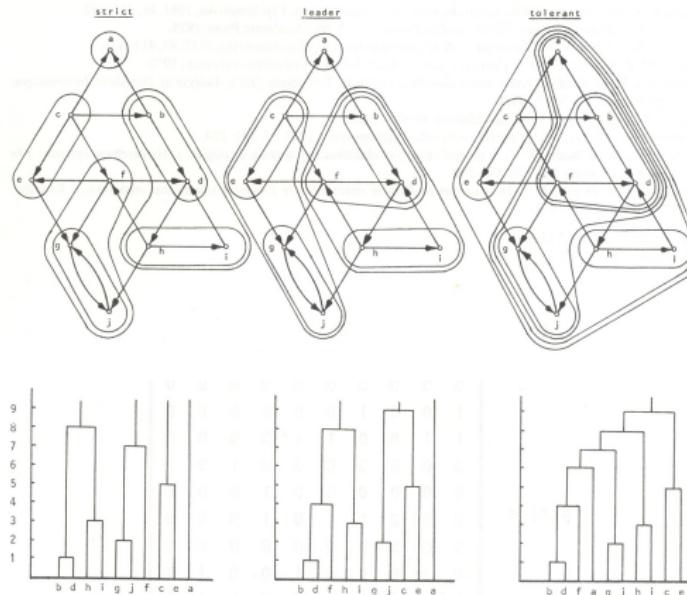
Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints





Dissimilarities between clusters

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

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

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

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



Hierarchies

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

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



Level functions

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

A mapping $h : \mathcal{T} \rightarrow \mathbb{R}_0^+$ is a *level function* on \mathcal{T} iff

I1. $\forall X \in \mathcal{U} : h(\{X\}) = 0$

I2. $C_p \subseteq C_q \Rightarrow h(C_p) \leq h(C_q)$

A simple example of level function is $h(C) = \text{card}((C)) - 1$.

Every hierarchy / level function determines an ultrametric dissimilarity on \mathcal{U}

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

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

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

is a complete hierarchy, and $h(C) = \text{diam}(C)$ 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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

For a given dissimilarity d on the set of units \mathcal{U} and relational constraint R we define the *k nearest neighbors graph*

$$\mathbf{G}_{NN} = (\mathcal{U}, A)$$

$(X, Y) \in A \Leftrightarrow Y$ is selected among the nearest neighbors of X and $X R Y$

By setting for $(X, Y) \in A$ its value to $w((X, Y)) = d(X, Y)$ 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

Clustering

V. Batagelj

Clustering

Block modeling

Generalized Blockmodeling

Pre-specified blockmodeling

Two-mode blockmodeling

Signed graphs

Generalizations

Clustering with constraints

Let $\mathcal{N}_{NN} = (\mathcal{U}, A, w)$ be a nearest neighbor network. A pair of units $X, Y \in \mathcal{U}$ are *reciprocal nearest neighbors* or RNNs iff $(X, Y) \in A$ and $(Y, X) \in A$.

Suppose $\text{card}(\mathcal{U}) > 1$ and R has no isolated units. Then in \mathcal{N}

- every unit/vertex $X \in \mathcal{U}$ has the $\text{outdeg}(X) \geq 1$ — 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 $\min_{X, Y \in \mathcal{U}, X \neq Y} d(X, Y)$.



Fast agglomerative clustering algorithms

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

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 := 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 := Y ∪ Z; W := W \ {Y, Z} ∪ {T}; compute D(T, W), W ∈ 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

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

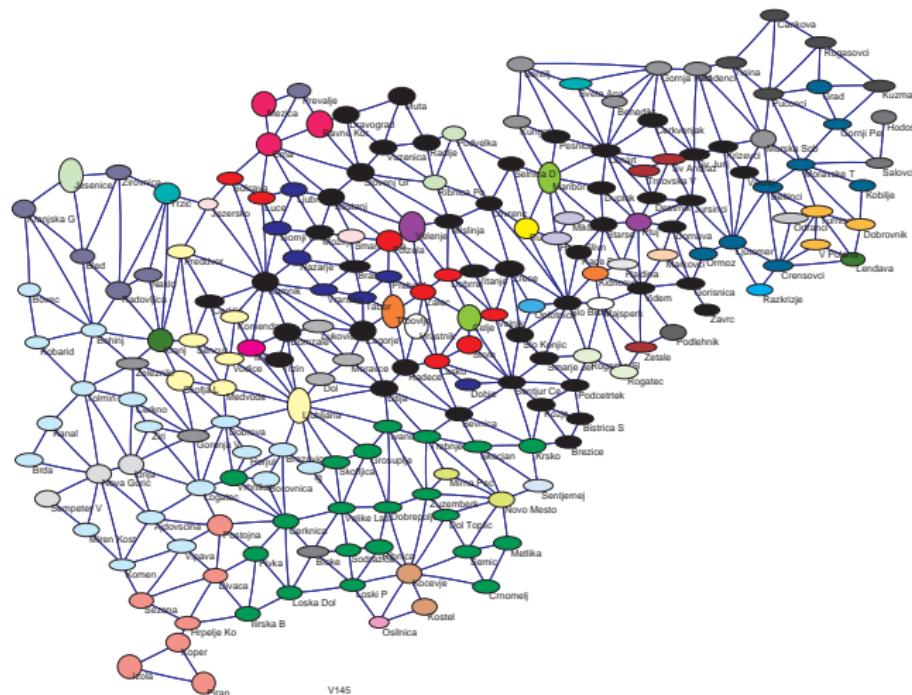
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Example: US counties $t = 1400$

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

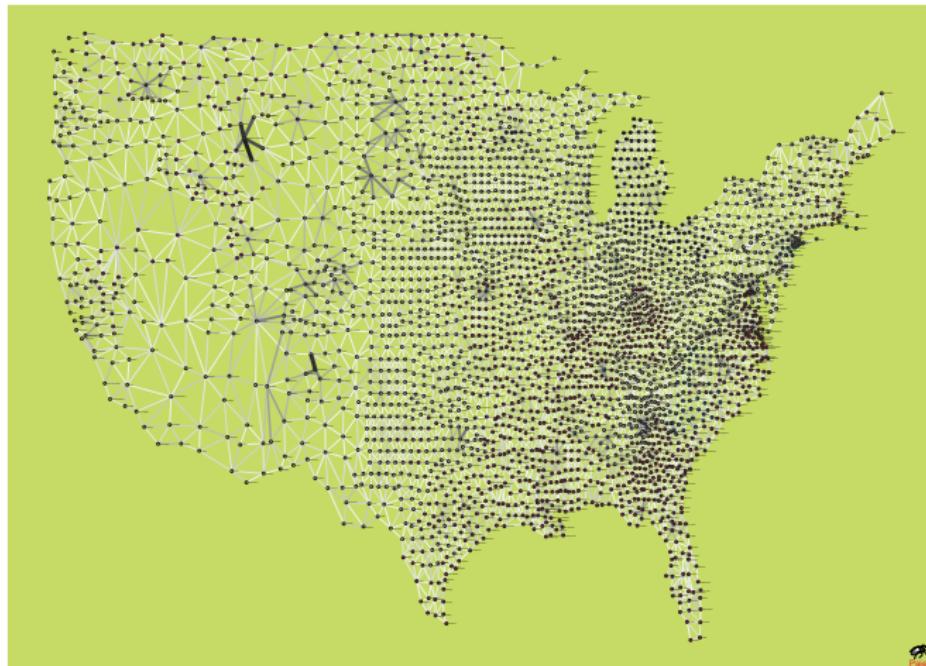
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Example: US counties $t = 200$

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

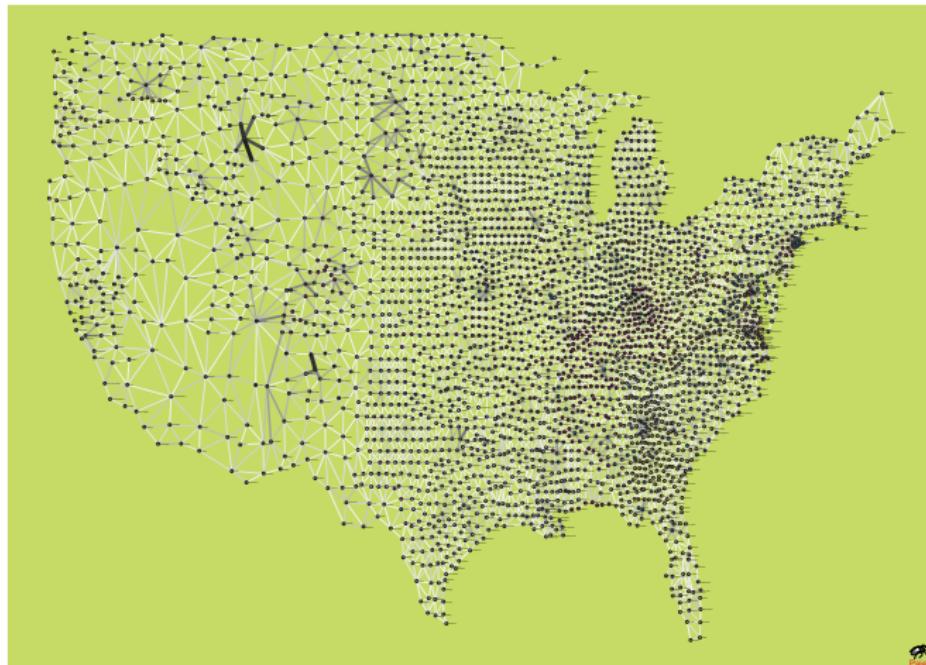
Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints





Louvain method and VOS

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodeling

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

One of the approaches to determine the clustering \mathbf{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 nodes belonging to cluster C , and $d(C)$ is the sum of the degrees of nodes from C .

The modularity maximization problem is NP complete.

Louvain method and VOS.

Network/Create Partition/Communities/Louvain Method

Network/Create Partition/Communities/VOS Clustering

[Draw] Layout/VOS Mapping



References

Clustering

V. Batagelj

Clustering

Block
modeling

Generalized
Blockmodel-
ing

Pre-specified
blockmodeling

Two-mode
blockmodeling

Signed graphs

Generalizations

Clustering
with
constraints

- 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.