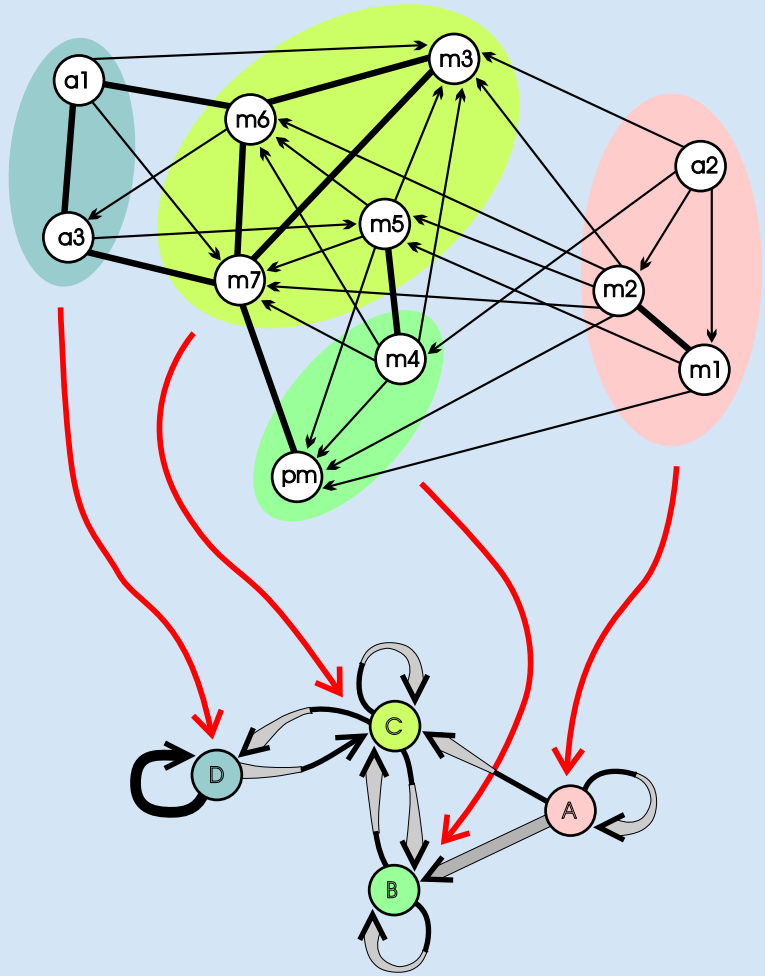


Clustering Relational Data

Anuška Ferligoj
Faculty of Social Sciences
University of Ljubljana

version: January 20, 2009 / 08 : 35

Outline

Introduction

A large class of *clustering problems* can be formulated as an *optimizational problem* in which the best clustering is searched among all *feasible clusterings* according to a selected *criterion function*.

This clustering approach can be applied to a variety of very interesting clustering problems, as it is possible to adapt it to a concrete clustering problem by an appropriate specification of the criterion function and/or by the definition of the set of feasible clusterings.

Both, the *blockmodeling problem* (clustering of the relational data) and the *clustering with relational constraint problem* (clustering of the attribute and relational data) can be very successfully treated by this approach.

Cluster Analysis

Grouping units into clusters so that those within a cluster are as similar to each other as possible, while units in different clusters as dissimilar as possible, is a very old problem.

Although the clustering problem is intuitively simple and understandable, providing solution(s) remains a very exciting activity.

The field of cluster analysis

- has its society, the *International Federation of Classification Societies*, formed in 1985 from several national classification societies;
- organizes every second year its conference;
- publishes two journals: the *Journal of Classification* (from 1984) and the journal *Advances in Data Analysis and Classification* (from 2007).

Clustering Problem

Cluster analysis (known also as classification and taxonomy) deals mainly with the following general problem: given a set of *units*, $\mathcal{U}$, determine subsets, called *clusters*, C, which are homogeneous and/or well separated according to the measured variables. The set of clusters forms a *clustering*.

This problem can be formulated as an *optimization problem*:

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

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

where $\mathbf{C}$ is a clustering of a given set of units, $\mathcal{U}$, Φ is the set of all *feasible clusterings* and $P : \Phi \rightarrow R$ is a *criterion function*.

Clustering

There are several types of clusterings, e.g., partition, hierarchy, pyramid, fuzzy clustering, clustering with overlaping clusters. The most frequently used clusterings are partitions and hierarchies.

A clustering $\mathbf{C} = \{C_1, C_2, ...C_k\}$ is a *partition* of the set of units $\mathcal{U}$ if

$$\bigcup_i C_i = \mathcal{U}$$

$$i \neq j \Rightarrow C_i \cap C_j = \emptyset$$

A clustering $\mathbf{H} = \{C_1, C_2, ...C_k\}$ is a *hierarchy* if for each pair of clusters C_i and C_j from $\mathbf{H}$ it holds

$$C_i \cap C_j \in \{C_i, C_j, \emptyset\}$$

and it is a *complete hierarchy* if for each unit x it holds $\{x\} \in \mathbf{H}$, and $\mathcal{U} \in \mathbf{H}$.

Clustering Criterion Function

Clustering criterion functions can be constructed *indirectly*, e.g., as a function of a suitable (dis)similarity measure between pairs of units (e.g., euclidean distance) or *directly*.

For partitions into k clusters, the *Ward criterion function*

$$P(\mathbf{C}) = \sum_{C \in \mathbf{C}} \sum_{x \in C} d(x, t_C)$$

is usually used, where t_C is the center of the cluster C and is defined as

$$t_C = (\overline{u}_{1C}, \overline{u}_{2C}, ..., \overline{u}_{mC})$$

where $\overline{u}_{iC}$ is the average of the variable U_i, $i = 1, ...m$, for the units from the cluster C. d is the squared euclidean distance.

As the set of feasible clusterings is finite a solution of the clustering problem always exists. Since this set is usually very large it is not easy to find an optimal solution.

In general, most of the clustering problems are *NP-hard*. For this reason, different efficient *heuristic* algorithms are used. Among these, the *agglomerative* (hierarchical) and the *relocation* approach are most often used.

Agglomerative Approach

The agglomerative clustering approach usually assumes that all relevant information on the relationships between the n units from the set $\mathcal{U}$ is summarized by a symmetric pairwise *dissimilarity matrix* $D = [d_{ij}]$.

Each unit is a cluster: $C_i = \{x_i\}$, $x_i \in \mathcal{U}$, $i = 1, 2, \ldots, n$;
repeat while there exist at least two clusters:

determine the nearest pair of clusters C_p and C_q:

$d(C_p, C_q) = \min_{u,v} d(C_u, C_v)$;

fuse the clusters C_p and C_q to form a new cluster $C_r = C_p \cup C_q$;

replace C_p and C_q by the cluster C_r;

determine the dissimilarities between the cluster C_r and other clusters.

The result is a hierarchy that is usually presented by a clustering tree – a *dendrogram*.

Relocation Approach

This approach assumes that the user can specify the number of clusters in the partition.

Determine the initial clustering $\mathbf{C}$;
while
there exists $\mathbf{C}'$ such that $P(\mathbf{C}') \leq P(\mathbf{C})$, where $\mathbf{C}'$ is obtained
by moving a unit x_i from cluster C_p to cluster C_q,
or by interchanging units x_i and x_j between two clusters
in the clustering $\mathbf{C}$;
repeat:
substitute $\mathbf{C}'$ for $\mathbf{C}$.

Benefits from the Optimizational Approach

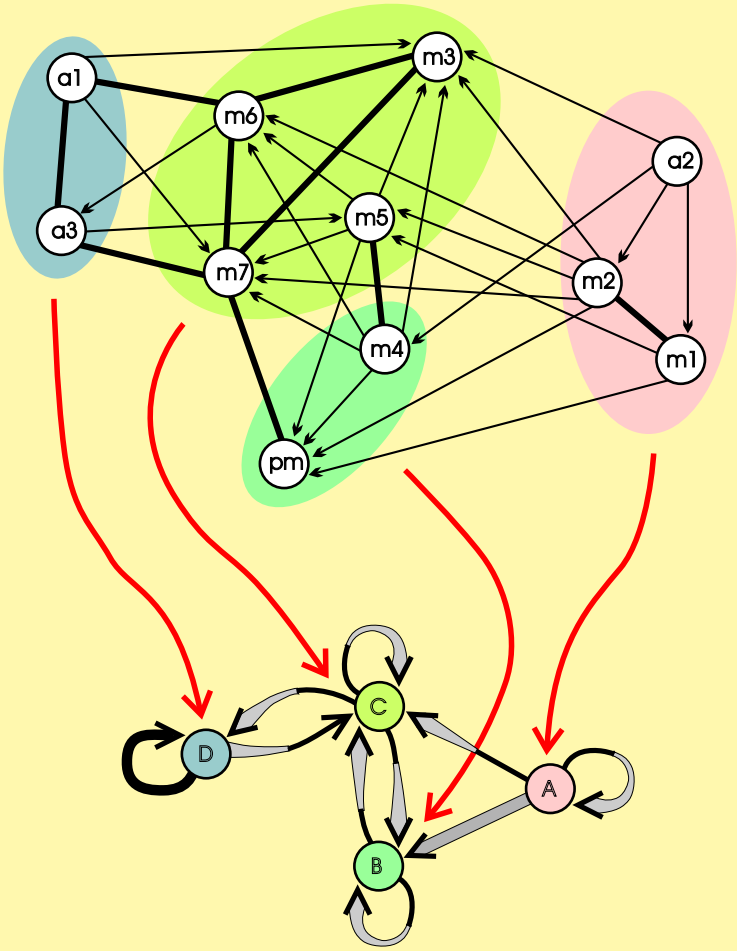
The *optimizational approch* to clustering problem offers two possibilities to adapt to a concrete clustering problem: the definition of the *criterion function* P and the specification of the *set of feasible clusterings* Φ.

Blockmodeling is searching for a clustering according to the *relational data only* and the solution can be obtained by an appropriatelly defined *criterion function*.

For *clustering with relational constraint* an appropriately defined set of *feasible clusterings* is used.

Blockmodeling

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

One of the main procedural goals of blockmodeling is to identify, in a given network $\mathbf{N} = (\mathcal{U}, R)$, $R \subseteq \mathcal{U} \times \mathcal{U}$, *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* $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ 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 $\mathbf{C}$.

A clustering $\mathbf{C}$ 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.

The Everett Network

	a	b	c	d	e	f	g	h	i	j
a	0	1	1	1	0	0	0	0	0	0
b	1	0	1	0	1	0	0	0	0	0
c	1	1	0	1	0	0	0	0	0	0
d	1	0	1	0	1	0	0	0	0	0
e	0	1	0	1	0	1	0	0	0	0
f	0	0	0	0	1	0	1	0	1	0
g	0	0	0	0	0	1	0	1	0	1
h	0	0	0	0	0	0	1	0	1	1
i	0	0	0	0	0	1	0	1	0	1
j	0	0	0	0	0	0	1	1	1	0

	a	c	h	j	b	d	g	i	e	f
a	0	1	0	0	1	1	0	0	0	0
c	1	0	0	0	1	1	0	0	0	0
h	0	0	0	1	0	0	1	1	0	0
j	0	0	1	0	0	0	1	1	0	0
b	1	1	0	0	0	0	0	0	1	0
d	1	1	0	0	0	0	0	0	1	0
g	0	0	1	1	0	0	0	0	0	1
i	0	0	1	1	0	0	0	0	0	1
e	0	0	0	0	1	1	0	0	0	1
f	0	0	0	0	0	0	1	1	1	0

	A	B	C
A	1	1	0
B	1	0	1
C	0	1	1

Equivalences

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

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

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

s1. $\mathrm{X}R\mathrm{Y} \Leftrightarrow \mathrm{Y}R\mathrm{X}$

s2. $\mathrm{X}R\mathrm{X} \Leftrightarrow \mathrm{Y}R\mathrm{Y}$

s3. $\forall \mathrm{Z} \in \mathcal{U} \setminus \{\mathrm{X}, \mathrm{Y}\} : (\mathrm{X}R\mathrm{Z} \Leftrightarrow \mathrm{Y}R\mathrm{Z})$

s4. $\forall \mathrm{Z} \in \mathcal{U} \setminus \{\mathrm{X}, \mathrm{Y}\} : (\mathrm{Z}R\mathrm{X} \Leftrightarrow \mathrm{Z}R\mathrm{Y})$

. . . Structural Equivalence

The blocks for structural equivalence are null or complete with variations on diagonal in diagonal blocks.

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

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

,

Regular Equivalence

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 $\mathbf{N} = (\mathcal{U}, R)$ if and only if for all $\mathrm{X}, \mathrm{Y}, \mathrm{Z} \in \mathcal{U}$, $\mathrm{X} \approx \mathrm{Y}$ implies both

R1. $\mathrm{X}R\mathrm{Z} \Rightarrow \exists \mathrm{W} \in \mathcal{U} : (\mathrm{Y}R\mathrm{W} \wedge \mathrm{W} \approx \mathrm{Z})$

R2. $\mathrm{Z}R\mathrm{X} \Rightarrow \exists \mathrm{W} \in \mathcal{U} : (\mathrm{W}R\mathrm{Y} \wedge \mathrm{W} \approx \mathrm{Z})$

. . . Regular Equivalence

Theorem 1 (Batagelj, Doreian, Ferligoj, 1992) *Let* $\mathbf{C} = \{C_i\}$ *be a partition corresponding to a regular equivalence* $\approx$ *on the network* $\mathbf{N} = (\mathcal{U}, R)$. *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

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

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

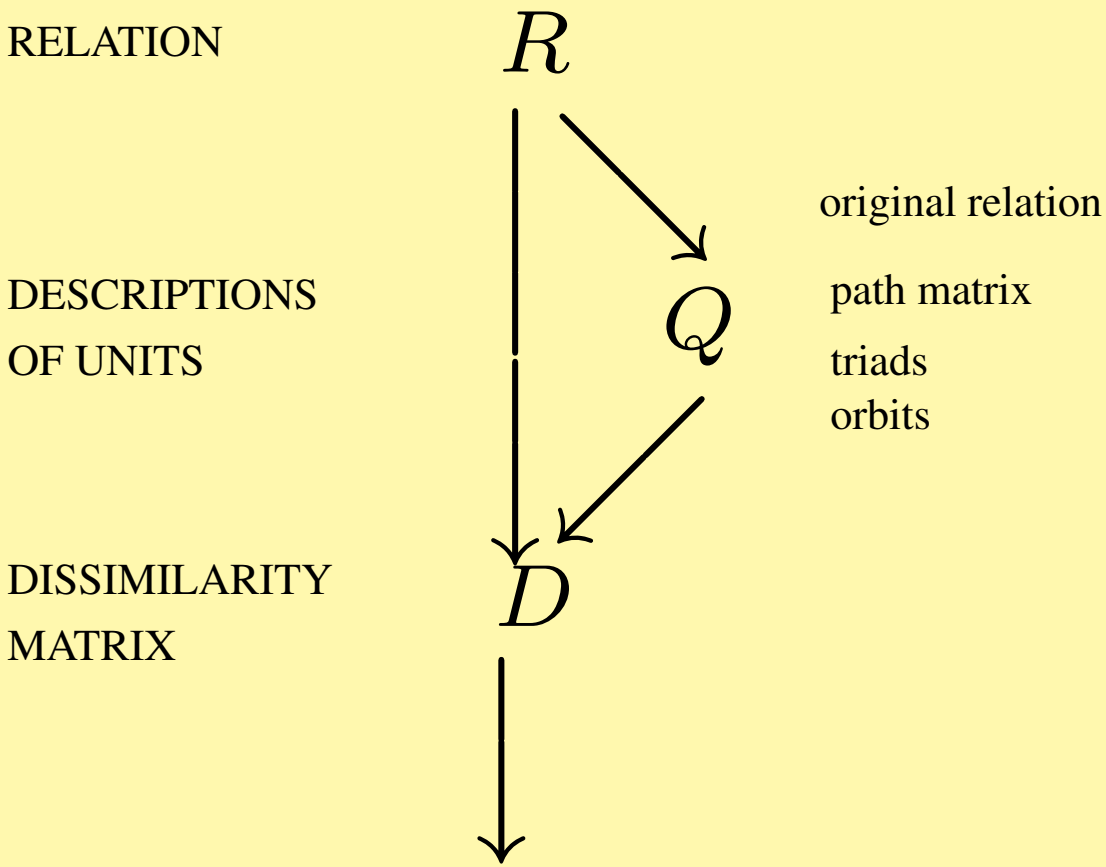
where Φ is the set of *feasible clusterings* and P is a *criterion function*.

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



Dissimilarities

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*

$$d(X_i, X_j) = \sqrt{(r_{ii} - r_{jj})^2 + (r_{ij} - r_{ji})^2 + \sum_{\substack{s=1 \\ s \neq i,j}}^{n} ((r_{is} - r_{js})^2 + (r_{si} - r_{sj})^2)}$$

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.

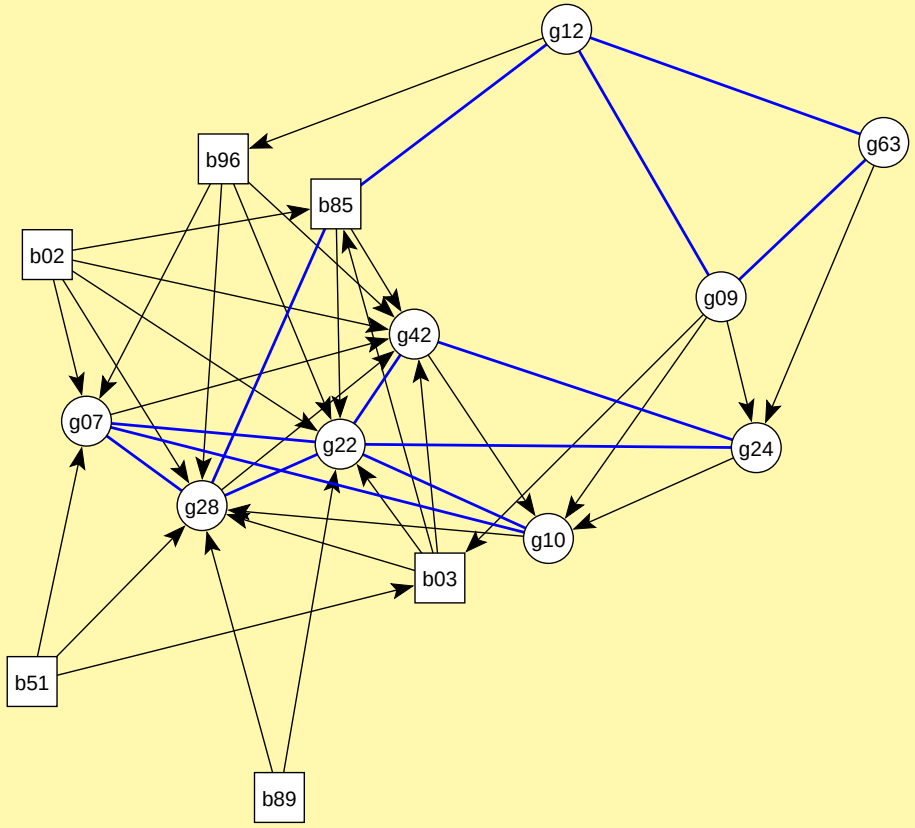
Example: Support Network among Informatics Students

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.)

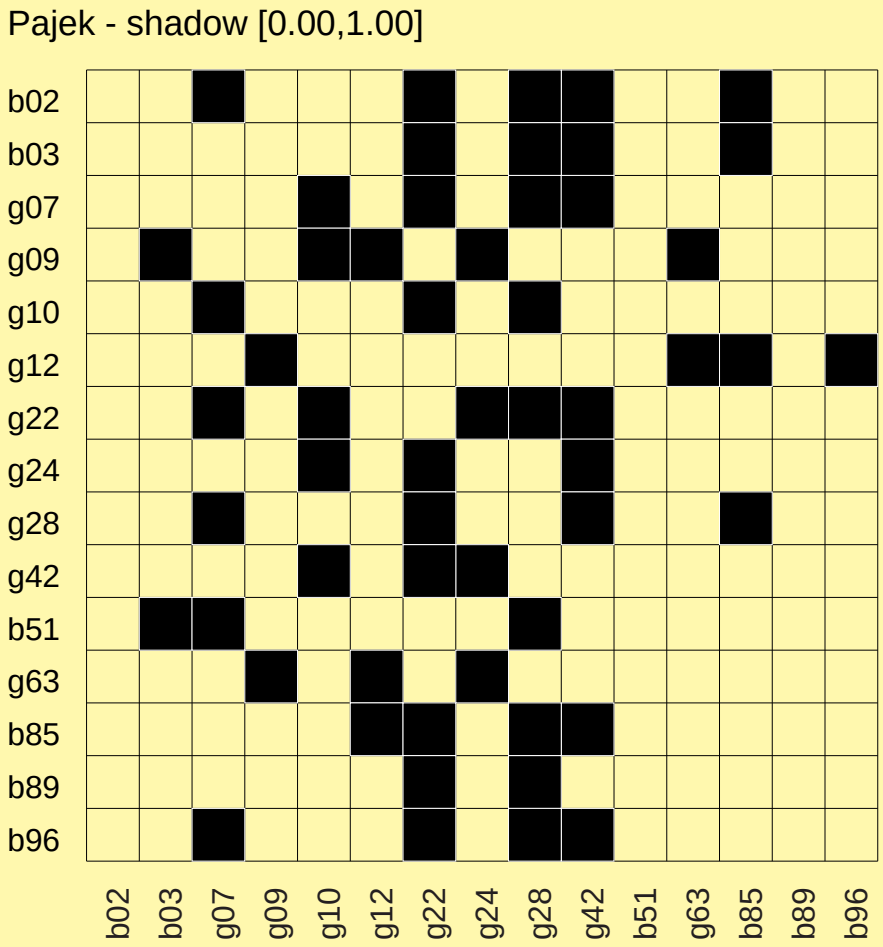
Class Network - Graph



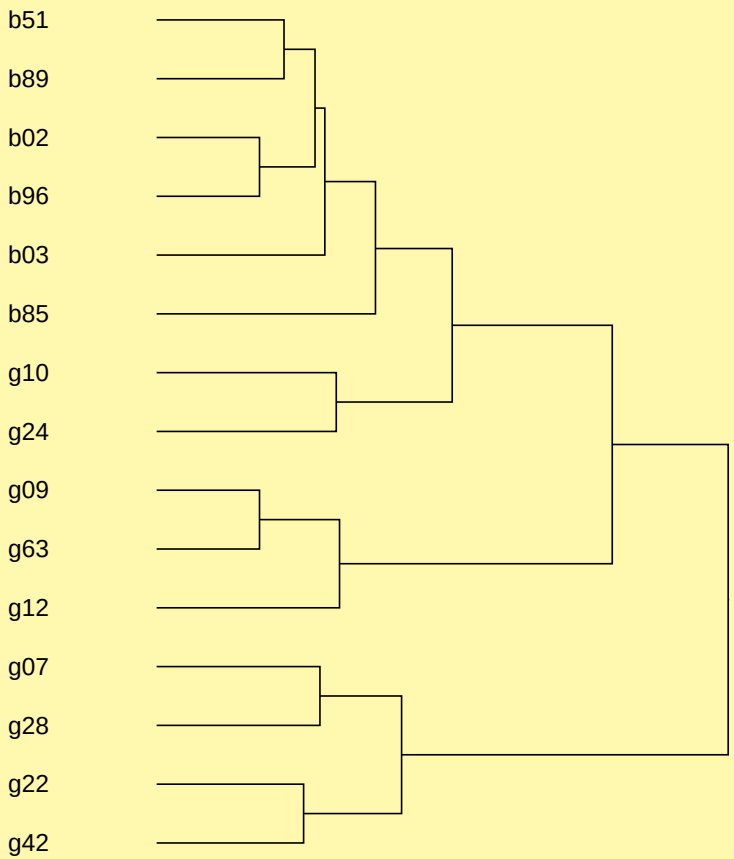
`class.net`

Vertices represent students in the class; circles – girls, squares – boys. Reciprocated arcs are represented by edges.

Class Network – Matrix



Indirect Approach



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

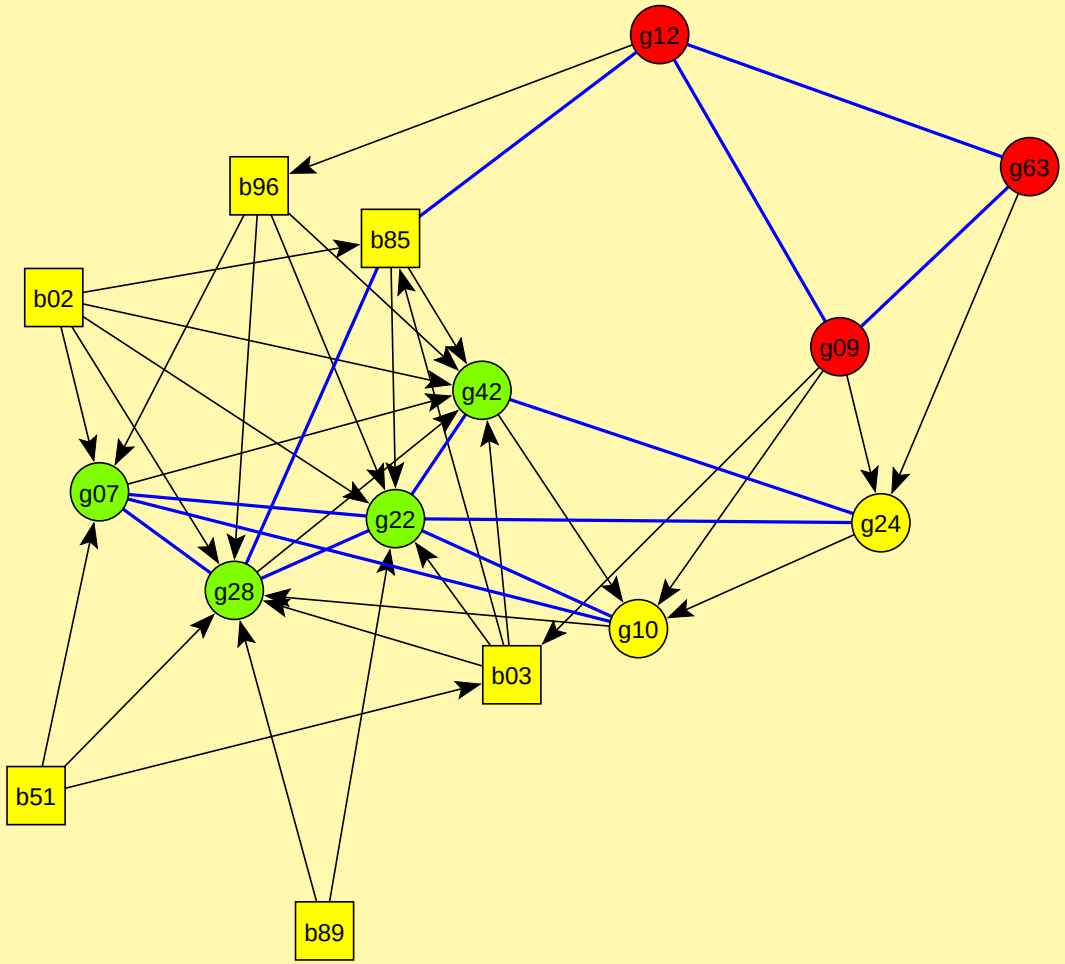
From it we can determine the number of clusters: 'Natural' clusterings correspond to clear 'jumps' in the dendrogram.

If we select 3 clusters we get the partition $\mathbf{C}$.

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

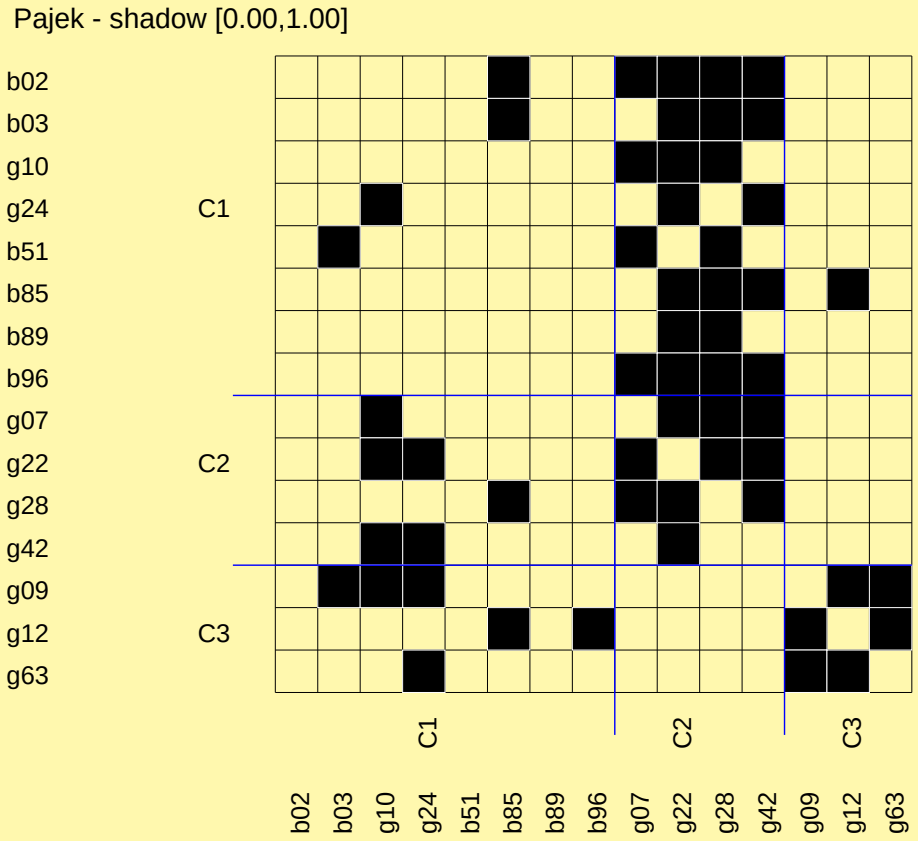
,

Partition into Three Clusters (Indirect Approach)



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.

,

Direct Approach

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:

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

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.

Empirical blocks

	a	b	c	d	e	f	g
a	0	1	1	0	1	0	0
b	1	0	1	0	0	0	0
c	1	1	0	0	0	0	0
d	1	1	1	0	0	0	0
e	1	1	1	0	0	0	0
f	1	1	1	0	1	0	1
g	0	1	1	0	0	0	0

Ideal blocks

	a	b	c	d	e	f	g
a	0	1	1	0	0	0	0
b	1	0	1	0	0	0	0
c	1	1	0	0	0	0	0
d	1	1	1	0	0	0	0
e	1	1	1	0	0	0	0
f	1	1	1	0	0	0	0
g	1	1	1	0	0	0	0

Number of inconsistencies for each block

	A	B
A	0	1
B	1	2

The value of the criterion function is the sum of all inconsistencies $P = 4$.

Local Optimization

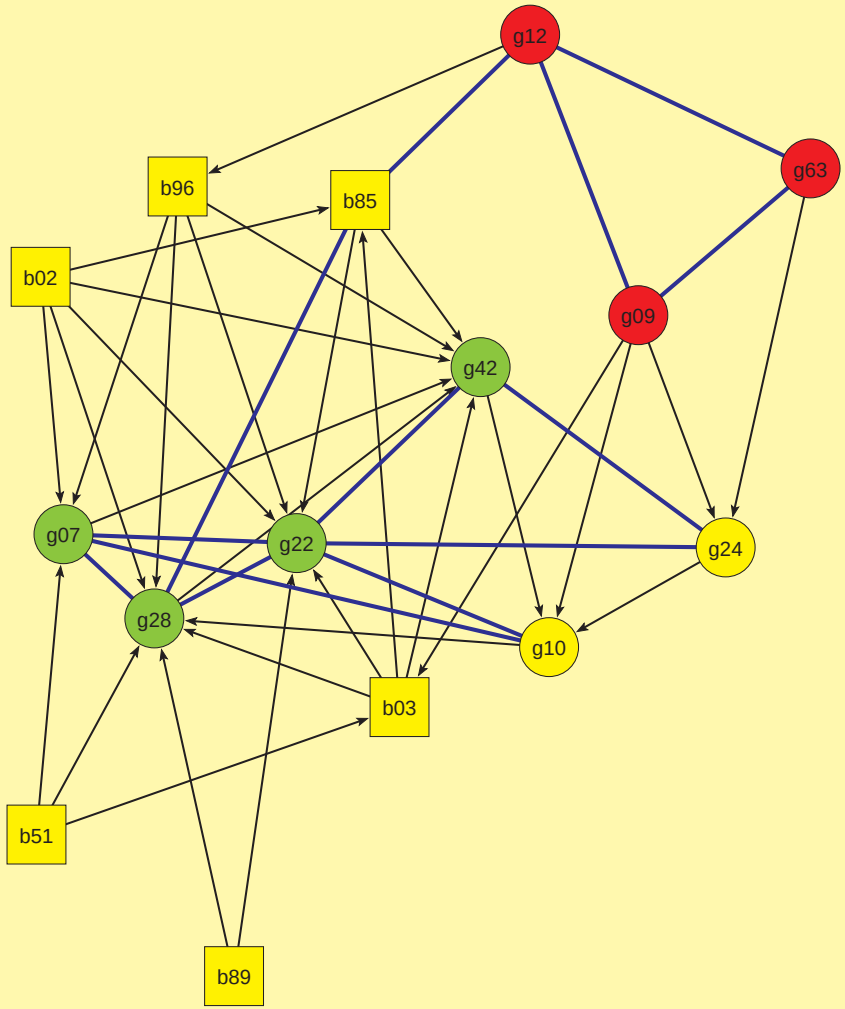
For solving the blockmodeling problem we use the relocation algorithm:

Determine the initial clustering $\mathcal{C}$;
repeat:
if in the neighborhood of the current clustering $\mathcal{C}$
there exists a clustering $\mathcal{C}'$ such that $P(\mathcal{C}') < P(\mathcal{C})$
then move to clustering $\mathcal{C}'$.

The neighborhood in this local optimization procedure is determined by the following two transformations:

- *moving* a unit X_k from cluster C_p to cluster C_q (*transition*);
- *interchanging* units X_u and X_v from different clusters C_p and C_q (*transposition*).

Partition into Three Clusters: Direct Solution (Unique)

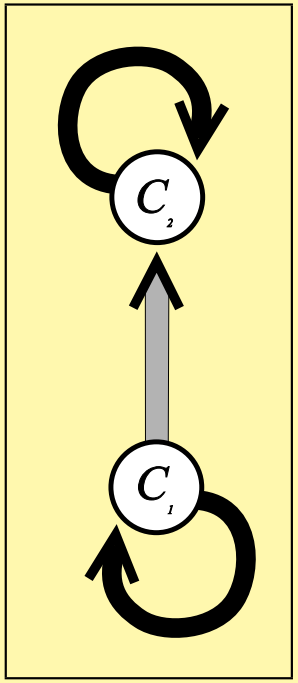


This is the same partition and has the number of inconsistencies.

Generalized Blockmodeling

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



Generalized Equivalence / Block Types

	Y				
X	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1

complete

	Y				
X	0	1	0	0	0
	1	1	1	1	1
	0	0	0	0	0
	0	0	0	1	0

row-dominant

	Y				
X	0	0	1	0	0
	0	0	1	1	0
	1	1	1	0	0
	0	0	1	0	1

col-dominant

	Y				
X	0	1	0	0	0
	1	0	1	1	0
	0	0	1	0	1
	1	1	0	0	0

regular

	Y				
X	0	1	0	0	0
	0	1	1	0	0
	1	0	1	0	0
	0	1	0	0	1

row-regular

	Y				
X	0	1	0	1	0
	1	0	1	0	0
	1	1	0	1	1
	0	0	0	0	0

col-regular

	Y				
X	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0

null

	Y				
X	0	0	0	1	0
	0	0	1	0	0
	1	0	0	0	0
	0	0	0	1	0

row-functional

	Y				
X	1	0	0	0	
	0	1	0	0	
	0	0	1	0	
	0	0	0	0	
	0	0	0	1	

col-functional

,

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:

*	*	*
*	0	0
*	0	0

center - periphery

*	0	0
*	*	0
?	*	*

hierarchy

*	0	0
0	*	0
0	0	*

clustering

Prespecified Blockmodeling Example

We expect that center-periphery model exists in the network: some students having good studying material, some not.

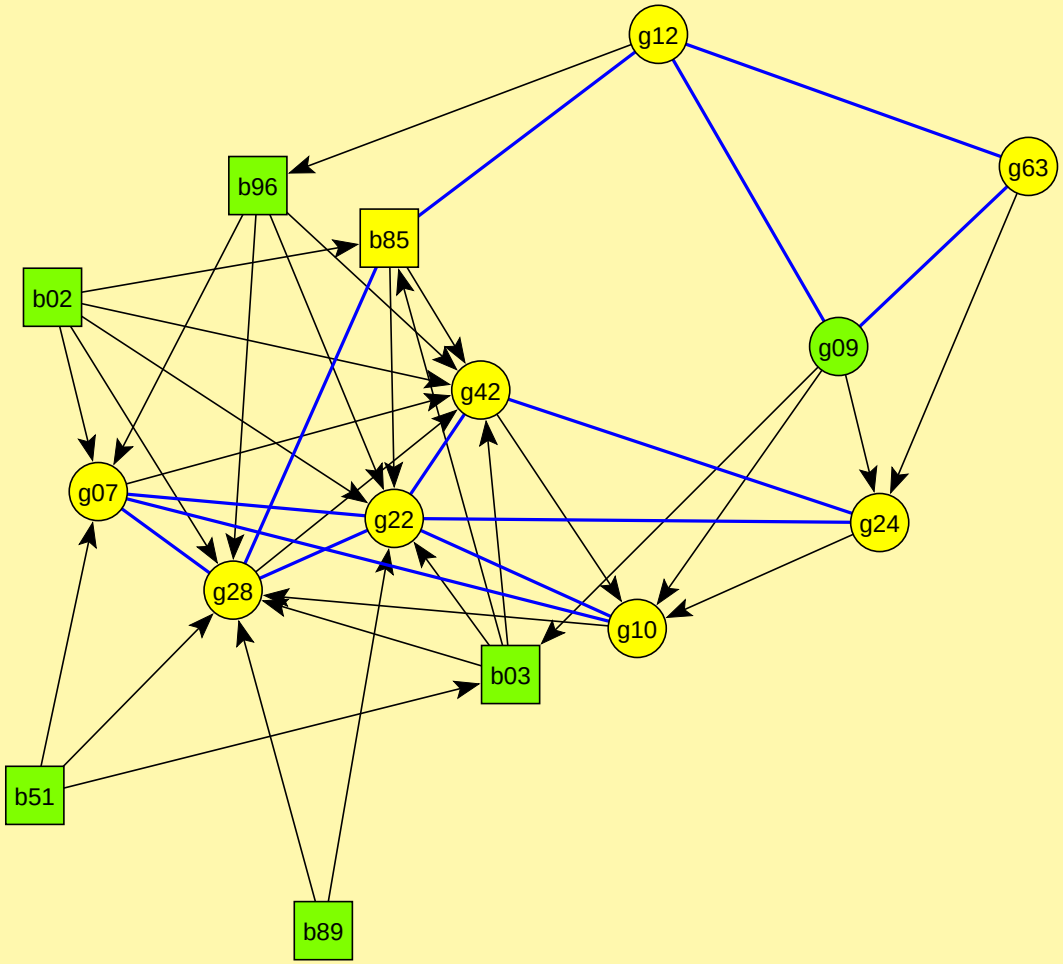
Prespecified blockmodel: (com/complete, reg/regular, -/null block)

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

Using local optimization we get the partition:

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

2 Clusters Solution



,

Model

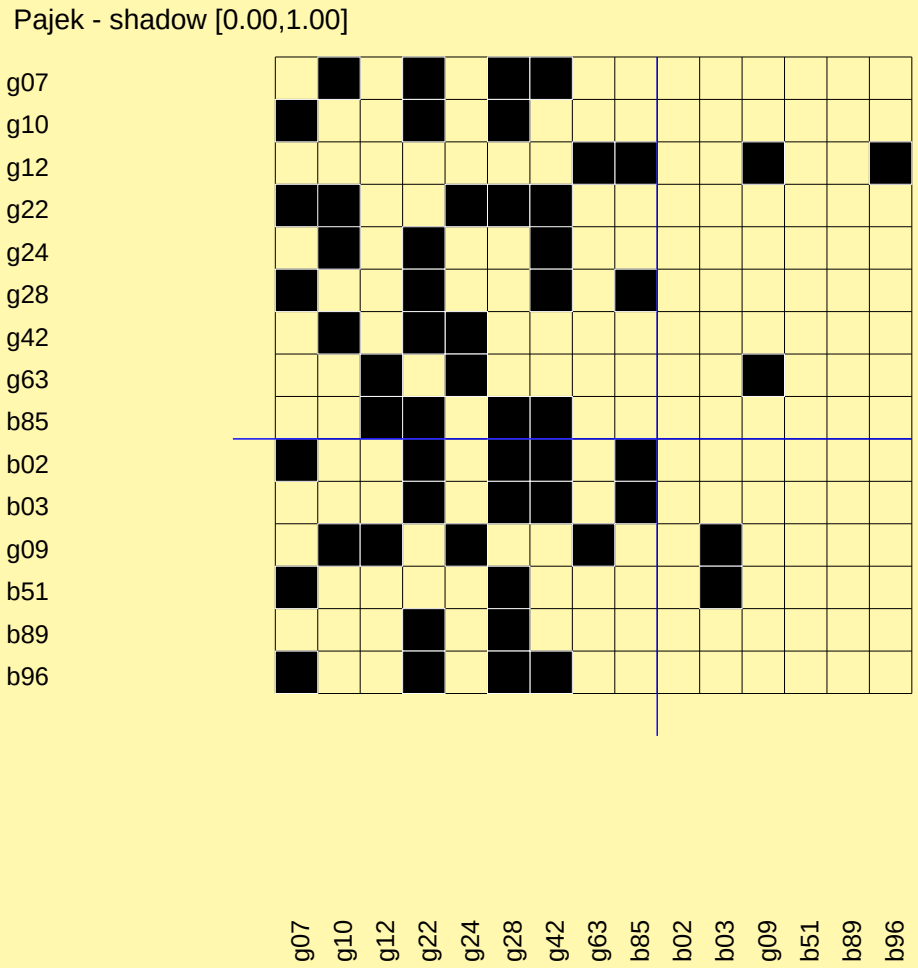


Image and Error Matrices:

	1	2
1	reg	-
2	reg	-

	1	2
1	0	3
2	0	2

Total error = 5

center-periphery

,

Blockmodeling of Multi-Way Network

It is also possible to formulate a generalized blockmodeling problem where the network is defined by several sets of units and ties between them. Therefore, several partitions – for each set of units a partition has to be determined. The generalized blockmodeling approach was adapted for 2-way networks, and only for structural equivalence and the indirect approach for 3-way networks.

Blockmodeling of Valued Networks

Till now we were treating only binary networks. Another interesting problem is the development of generalized blockmodeling of valued networks. Žiberna (2007) proposed several approaches to generalized blockmodeling of valued networks, where values of the ties are assumed to be measured on at least interval scale.

Clustering with Relational Constraints

Departements and regions of France

To group given territorial units into regions such that units inside the region will be similar according to selected *variables* (attributes) and form *contiguous* part of the territory was the motivation to develop *clustering with relational constraints approach* (Ferligoj and Batagelj, 1982 and 1983).

,

. . . Clustering with Relational Constraint

In the case of clustering with the relational constraint, the problem is to find clusterings as similar as possible according to attribute data and also considering the ties from a relation R. The constrained clustering problem can be expressed as clustering problem where the constraints are considered in the definition of the feasible clusterings.

The clustering with constraints problem seeks to determine the clustering $\mathbf{C}^*$ for which the criterion function P has the minimal value among all clusterings from the set of feasible clusterings $\mathbf{C} \in \Phi$, where Φ *is determined by the constraints*.

Set of feasible clusterings for relational type of constraint can be defined as:

$$\begin{aligned}\Phi(R) = \{&\mathbf{C} : \mathbf{C} \text{ is a partition of } \mathcal{U} \text{ and each cluster } C \in \mathbf{C} \\ &\text{is a subgraph } (C\ ,\ R \cap C \times C) \text{ in the graph } (\mathcal{U}, R) \\ &\text{with the required type of connectedness}\}\end{aligned}$$

. . . Clustering with Relational Constraints

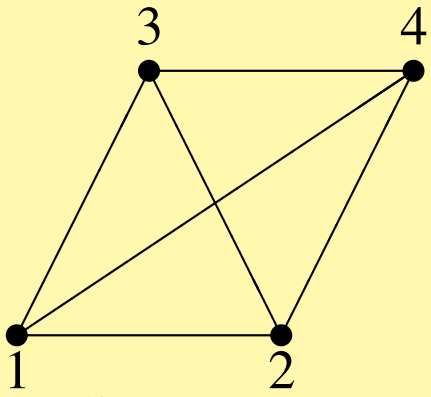
We can define different types of sets of feasible clusterings for the same relation R. Some examples of *types of relational constraint* $\Phi^i(R)$ are

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

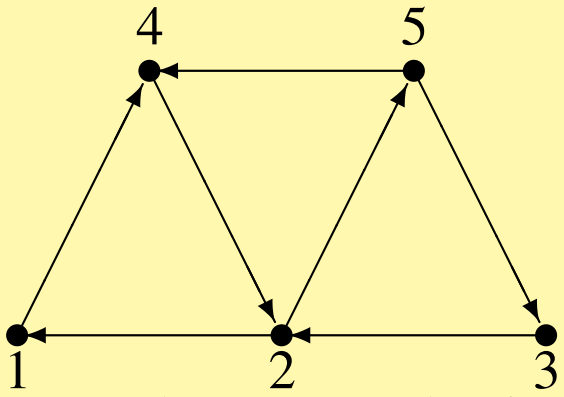
Trail – all arcs are distinct.

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 $R(L) \cap (C \setminus L) = \emptyset$.

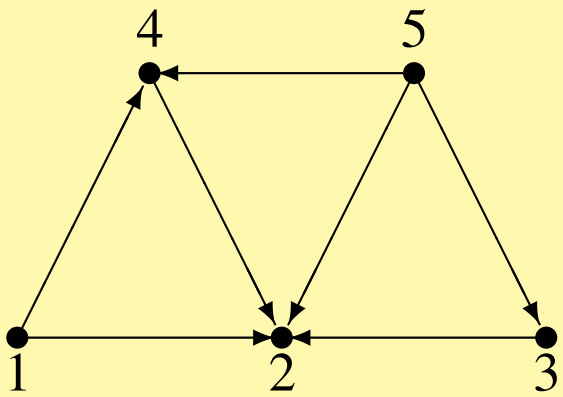
Some Graphs of Different Types



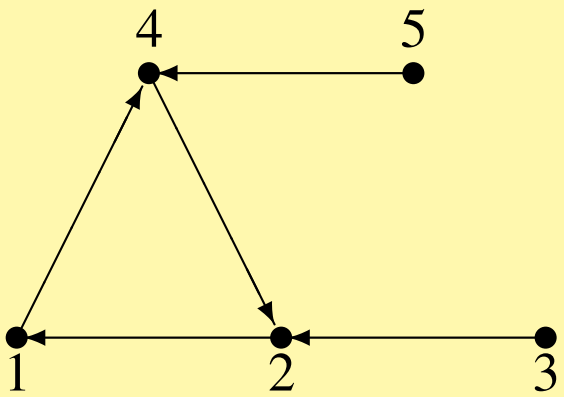
a clique



strongly connected units



weakly connected units



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

Solving Clustering with Relational Constraints Problem

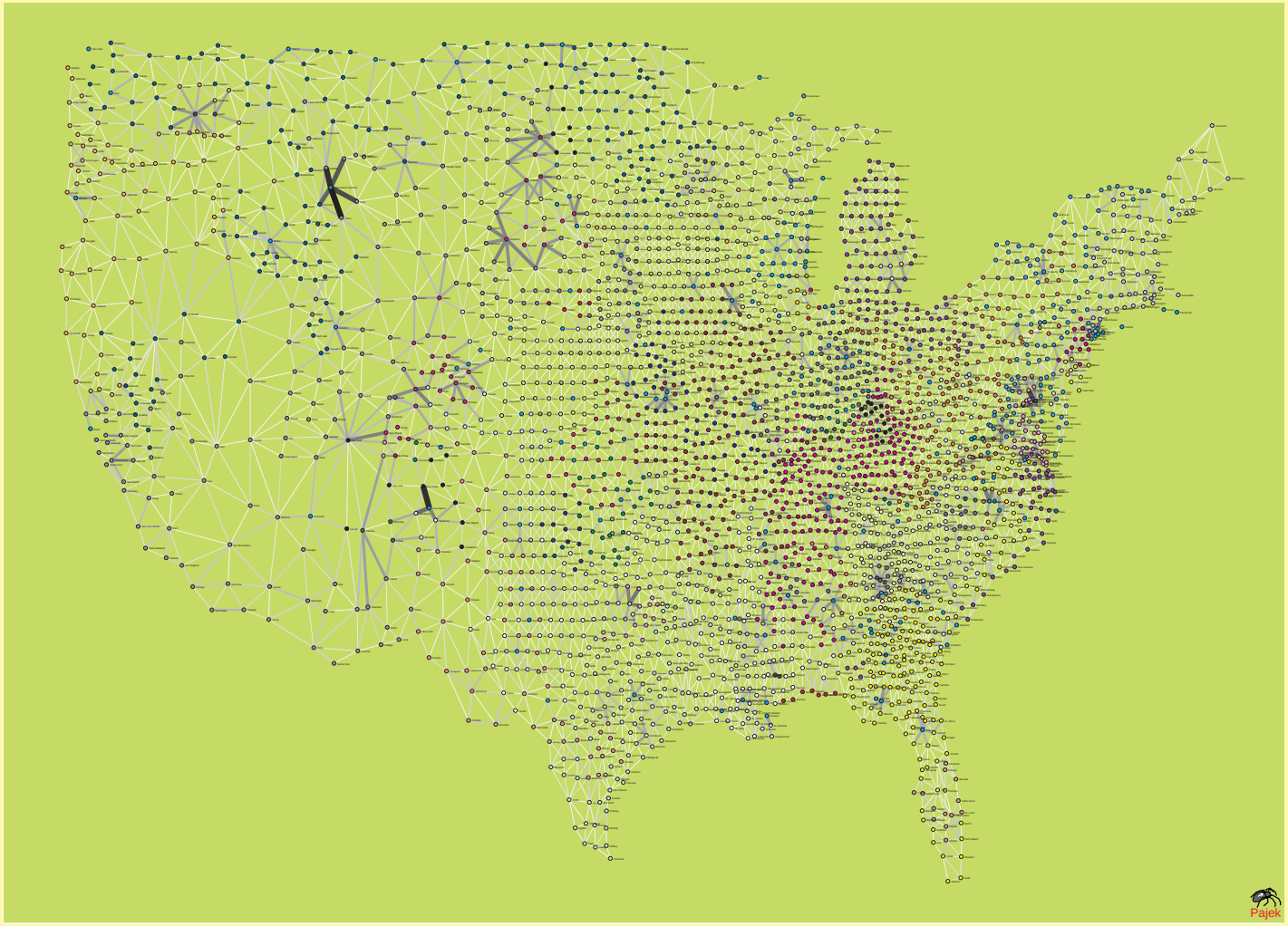
The agglomerative hierarchical and the relocation approach can be adapted for solving relational constrained clustering problems (Ferligoj and Batagelj, 1982 and 1983).

Clustering with Relational Constraints for Large Data

Recently Batagelj, Ferligoj and Mrvar (2007, 2008) adapted the clustering with relational constraint approach for very large networks. It is available in program **Pajek** .

Example: US Counties

US Census 2000: V1 – Area, V2 – Population, V47 – Percent of White, V125 – Educational attainment 1990, V126 – Household income; standardized



,

Software

Most of described clustering procedures are implemented in Pajek – program for analysis and visualization of large networks (Batagelj and Mrvar, 1998). It is freely available, for noncommercial use, at:

http://pajek.imfm.si

Conclusion

The optimizational approach to the clustering problem can be applied to a variety of very interesting clustering problems, as it alows possible adaptations of a concrete clustering problem by an appropriate specification of the criterion function and by the definition of the set of feasible clusterings. Both the blockmodeling problem and the clustering with relational constraint problem are such cases.

There are several possible further developments in blockmodeling, e.g., efficient direct approach for 3-way blockmodeling, blockmodeling for large networks, and dynamic blockmodels.