Notes on Blockmodeling*

Vladimir Batagelj
Department of Mathematics, University of Ljubljana
Jadranska 19, 61 111 Ljubljana
Slovenia

e–mail: vladimir.batagelj@uni-lj.si

First version: August 28, 1992
Last version: July 17, 1994

Abstract

In the paper an attempt to further develop the blockmodeling of networks to better capture the network structure is presented. For this purpose a richer structure than ordinary (valued) graphs has to be used for a model. Such structures are valued graphs with typified (complete, dominant, regular, ...) connections. Based on the proposed formalization the blockmodeling is expressed as an optimization problem.

Keywords: blockmodels, types of connection, averaging rules, optimization.

AMS Subj. Class. (1991): 92 H 30, 05 C 70, 90 C 27, 68 R 10.

1 Introduction

The paper is an elaboration of the following two basic observations:

- in blockmodeling we have two basic subproblems:
  - partitioning of units – determining the classes (clusters) that form the vertices in a model;
  - determining the links in a model (and their values);

*Extended version of the paper presented at the 3rd European Conference on Social Network Analysis, DJI, München, June 10-13, 1993.

- description of a model as a (valued) graph is often unprecise. For example, how to express the father–sons connection? A richer structure is needed for a model to be able to properly describe the network structure.

In this paper we propose a generalization of blockmodeling which enables us to better capture the network structure. It, among other, unifies and combines different notions of equivalences (structural, regular, ...), which can be simultaneously applied to the same network.

The paper deals mainly with technical (formal and computational) aspects of the proposed approach. A discussion of its methodological aspects and its relations to standard blockmodeling methods is given in [5].

2 Graphs and networks

Graph is an ordered triple $G = (V, E, A)$ where V, E and A are pairwise disjoint sets. The set V is a *vertex-set* of graph G; E is the set of *edges* (undirected lines), and A is the set of *arcs* (directed lines) of graph G. Sets E and A can also be empty. If $A = \emptyset$, the graph G is *undirected*; and it is *directed* if $E = \emptyset$. To each *line* from $L = E \cup A$ belongs a pair of vertices – its *ends*. In the case of an arc one vertex is its *initial* vertex, and the other vertex is its *terminal* vertex.

The fact, that the edge e has end vertices u and v we write $e(u{:}\, v)$; or equivalently $e(v{:}\, u)$. Similary $a(u, v)$ says that u is the initial and v is the terminal vertex of arc a. A line $p \in L$ *joins* its end vertices, and an arc $a \in A$ *joins* its initial vertex *to* its terminal vertex. When both ends of a line are equal we call it a *loop*. A vertex which is not an end of any line is called *isolated*.

We extend our notation for edges and arcs to all lines by: let $p \in L$, then

$$p(u, v) \equiv (p \in E \wedge p(u{:}\, v)) \vee (p \in A \wedge p(u, v))$$

line p joins vertex u to vertex v; and

$$p(u{:}\, v) \equiv p(u, v) \vee p(v, u)$$

line p joins vertices u and v.

We shall use the following abbreviation:

$$uLv \equiv \exists p \in L : p(u, v)$$

which essentially defines the *adjacency* relation. For $\emptyset \subset X, Y \subseteq V$ we also define a *block*

$$L(X, Y) = \{p \in L : \exists x \in X \exists y \in Y : p(x, y)\}$$

and a *complete block*

$$K(X, Y) = \{(x{:}\, y) : x \in X, y \in Y\} \cup \{(x, y) : x \in X, y \in Y\}$$

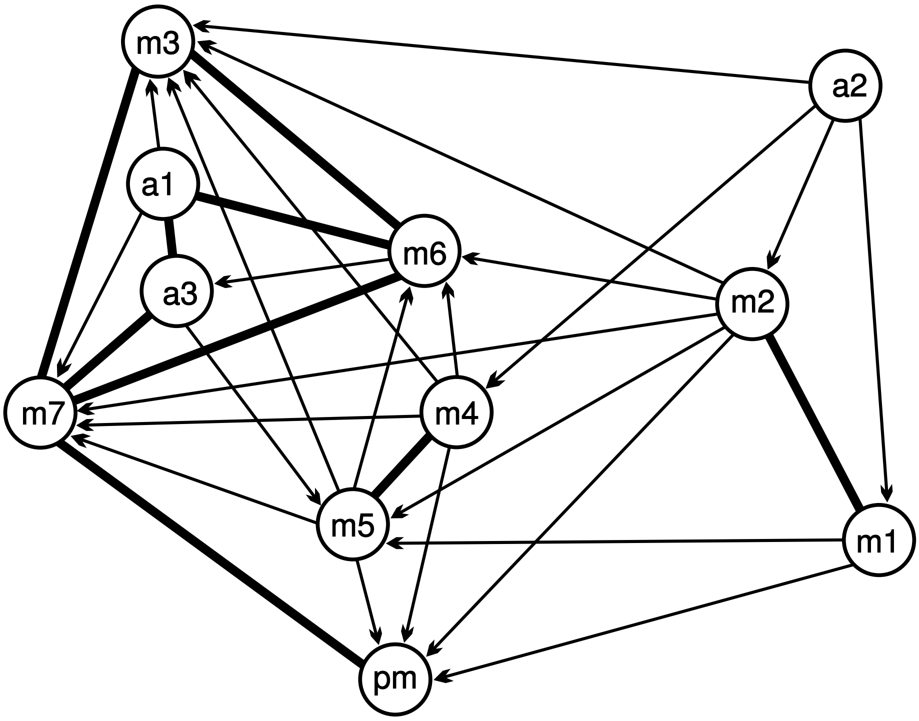

Figure 1: Network graph – discussion, recall

A block of the form $L(X, X)$ is called a *diagonal* block; and a block of the form $L(X, Y)$, $X \cap Y = \emptyset$, an *out-diagonal* block.

A graph is *simple* iff each pair of adjacent vertices is either joined by an edge, or by an arc, or by a pair of opposite arcs, or by a directed loop. In the following we shall assume all graphs simple.

A (simple) *network* is an ordered triple $\mathcal{N} = (V, L, \nu)$ where

- (V, L) is a graph,
- $\nu : L \to S$ assigns values to lines, S is the set of possible values.

In this context the set of vertices V is usually called the *set of units*. When the values of lines are not given we assume the 'default' values

$$\nu(x, y) = \begin{cases} 1 & xLy \\ 0 & \text{otherwise} \end{cases}$$

An example of a network is presented on Figure 1 [8]. The units are members of Student Government at University of Ljubljana in May 1992 (a – advisor, m – minister, pm – prime minister) and the relation is determined by the answers (recall) to the question:

Table 1: Types of connection

null	$\mathrm{nul}(X, Y; L)$	$\equiv$	$L(X, Y) = \emptyset$
complete	$\mathrm{com}(X, Y; L)$	$\equiv$	$\forall x \in X \forall y \in Y : (x \neq y \Rightarrow xLy)$
row-dominant	$\mathrm{rdo}(X, Y; L)$	$\equiv$	$\exists x \in X \forall y \in Y : (x \neq y \Rightarrow xLy)$
col-dominant	$\mathrm{cdo}(X, Y; L)$	$\equiv$	$\mathrm{rdo}(Y, X; L^{-1})$
row-regular	$\mathrm{rre}(X, Y; L)$	$\equiv$	$\forall x \in X \exists y \in Y : xLy$
col-regular	$\mathrm{cre}(X, Y; L)$	$\equiv$	$\mathrm{rre}(Y, X; L^{-1})$
regular	$\mathrm{reg}(X, Y; L)$	$\equiv$	$\mathrm{cre}(X, Y; L) \wedge \mathrm{rre}(X, Y; L)$
row-functional	$\mathrm{rfn}(X, Y; L)$	$\equiv$	$\forall y \in Y \exists! x \in X : xLy$
col-functional	$\mathrm{cfn}(X, Y; L)$	$\equiv$	$\forall x \in X \exists! y \in Y : xLy$

Who of the members and advisors of the Student Government do you (most often) informally discuss with?

3 Types of connection

Two sets of vertices $X, Y \subseteq V$ can be related in different ways between them. We describe these *types* of connection by predicates, where X is considered as the ego-set (see Table 1 and Figure 2).

These predicates have several characteristic properties

property	T
$T(X_1, Y_1) \wedge T(X_2, Y_2) \Rightarrow T(X_1 \cup X_2, Y_1 \cup Y_2)$	reg, rre, cre
$T(X_1, Y) \wedge T(X_2, Y) \Rightarrow T(X_1 \cup X_2, Y)$	com, rfn, nul, (reg, rre, cre)
$T(X, Y_1) \wedge T(X, Y_2) \Rightarrow T(X, Y_1 \cup Y_2)$	com, cfn, nul, (reg, rre, cre)
$\emptyset \subset Z \subseteq X \wedge T(X, Y) \Rightarrow T(Z, Y)$	com, nul, cdo, rfn, rre
$\emptyset \subset Z \subseteq Y \wedge T(X, Y) \Rightarrow T(X, Z)$	com, nul, rdo, cfn, cre
$T(X, Y) \Rightarrow T(X \cup Z, Y)$	rdo, cre
$T(X, Y) \Rightarrow T(X, Y \cup Z)$	cdo, rre

and several relations hold among them:

$$\begin{gathered}
\mathrm{reg}(X, Y) \Rightarrow \mathrm{cre}(X, Y), \qquad \mathrm{reg}(X, Y) \Rightarrow \mathrm{rre}(X, Y), \\
\mathrm{com}(X, Y) \Rightarrow \mathrm{rdo}(X, Y), \qquad \mathrm{rfn}(X, Y) \Rightarrow \mathrm{rre}(X, Y), \\
\mathrm{com}(X, Y) \Rightarrow \mathrm{cdo}(X, Y), \qquad \mathrm{cfn}(X, Y) \Rightarrow \mathrm{cre}(X, Y), \\
X \cap Y = \emptyset \wedge \mathrm{rdo}(X, Y) \Rightarrow \mathrm{cre}(X, Y), \\
X \cap Y = \emptyset \wedge \mathrm{cdo}(X, Y) \Rightarrow \mathrm{rre}(X, Y)
\end{gathered}$$

Often a selected type of connection is restricted to diagonal/out-diagonal blocks.

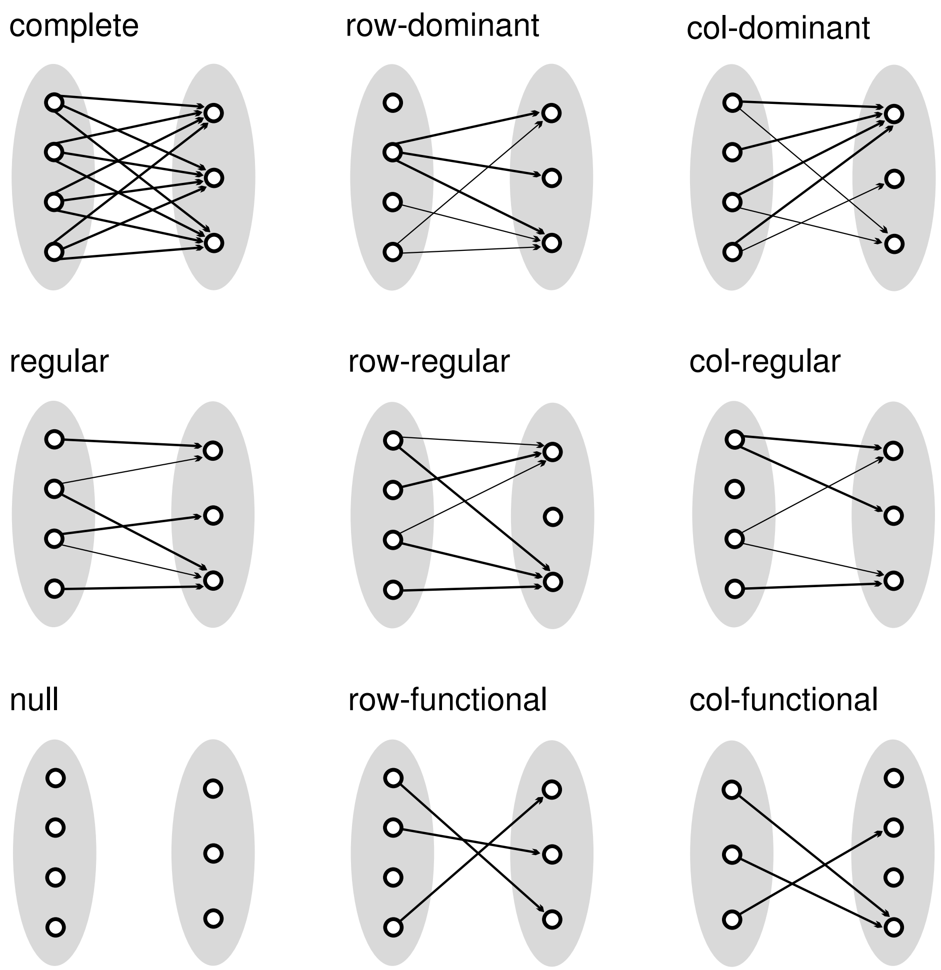

Figure 2: Types of connection

Another group of predicates is based on the notion of vertex degree. Examples of such predicates are:

degree density γ	$\mathrm{den}(\gamma)(X,Y;L)$	$\equiv$	$\mathrm{card}\, L(X,Y) \geq \gamma\, \mathrm{card}(X \times Y)$
degree bound n	$\mathrm{deg}(n)(X,Y;L)$	$\equiv$	$\forall x \in X : \mathrm{card}(L(x) \cap Y) \geq n$

More complicated predicates expressing partial ordering, different types of connectivity, ..., simultaneous consideration of $X \times Y$ and $Y \times X$, and even n-ary, $n > 2$ predicates could also be considered. In some applications a *don't care* predicate, which is always satisfied (true), can be useful. In this paper we shall limit our discussion only to binary predicates.

In the definition of predicates we can also consider values of lines in the block. For example, for searching balanced/clusterable partitions of a network two predicates are needed [6]:

positive	$\mathrm{pos}(X,Y;L)$	$\equiv$	$\forall x \in X, y \in Y : (xLy \Rightarrow \nu(x,y) > 0)$
negative	$\mathrm{neg}(X,Y;L)$	$\equiv$	$\forall x \in X, y \in Y : (xLy \Rightarrow \nu(x,y) < 0)$

4 Blockmodeling

4.1 Blockmodels

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

- U is a set of *types* of units (images or representatives of classes);
- $K \subseteq U \times U$ 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 $\mathrm{nul} \in \mathcal{T}$. A mapping $\pi : K \to \mathcal{T} \setminus \{\mathrm{nul}\}$ assigns predicates to connections;
- Q is a set of *averaging rules*. A mapping $\alpha : K \to Q$ determines rules for computing values of connections.

Let us denote by $\mu : V \to U$ a mapping which maps classes of units to the corresponding types. Then we define for $t \in U$

$$C(t) = \mu^{-1}(t) = \{x \in V : \mu(x) = t\}$$

Therefore

$$\mathcal{C}(\mu) = \{C(t) : t \in U\}$$

is a partition (clustering) of the set of units V.

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

$$\forall (t, w) \in K : \pi(t, w)(C(t), C(w))$$

and

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

Note that, if we set $\mathcal{T} = \{\text{nul}, \text{com}\}$ we are asking for a structural blockmodel [9]; and, if we set $\mathcal{T} = \{\text{nul}, \text{reg}\}$ we are asking for a regular blockmodel [11].

4.2 Equivalences

Let $\approx$ be an equivalence relation over V. It partitions the set of units V into classes (clusters)

$$[x] = \{y \in V : x \approx y\}$$

We say that $\approx$ is *compatible* with $\mathcal{T}$ over a network $\mathcal{N}$ iff

$$\forall x, y \in V \exists T \in \mathcal{T} : T([x], [y])$$

It is easy to verify that the notion of compatibility for $\mathcal{T} = \{\text{nul}, \text{reg}\}$ reduces to the usual definition of regular equivalence [4].

For a compatible equivalence $\approx$ the mapping $\mu : x \mapsto [x]$ determines a blockmodel.

4.3 Averaging rules

The next question is: How to determine the values of connections in a way compatible with their types and values of corresponding lines in a network? This can be done by selecting/determining an appropriate *averaging rule*.

Let for $t, w \in U$ be $X = C(t)$ and $Y = C(w)$, then general requirements for an averaging rule $\overline{\nu} : K \to S$ could be

$$\text{nul}(X, Y; L) \Rightarrow \overline{\nu}(t, w) = 0$$

and

$$(\forall p \in L(X, Y) : \nu(p) = c) \quad \Rightarrow \quad \overline{\nu}(t, w) = c$$

or

$$\sum_{p \in L(X,Y)} \nu(p) = N(t, w)\overline{\nu}(t, w)$$

where $N(t, w)$ is the *multiplicity* of connection (t, w). The multiplicity $N(t, w)$ depends also on the corresponding type of connection. For example, we can set:

$$\begin{aligned}
\text{com}(X, Y; L) &\Rightarrow N(t, w) = \text{card}(X \times Y) \\
\text{rre}(X, Y; L) &\Rightarrow N(t, w) = \text{card}\, Y \\
\text{reg}(X, Y; L) &\Rightarrow N(t, w) = \max(\text{card}\, X, \text{card}\, Y)
\end{aligned}$$

Table 2: Characterizations of types of blocks

null	all 0, (except may be diagonal)
complete	all 1, (except may be diagonal)
row-dominant	$\exists$ all 1 row, (except may be diagonal)
col-dominant	$\exists$ all 1 column, (except may be diagonal)
row-regular	1-covered rows
col-regular	1-covered columns
regular	1-covered rows $\wedge$ 1-covered columns
row-functional	exactly one 1 in each column
col-functional	exactly one 1 in each row
density γ	# of 1s $\geq \gamma \cdot size$

There are several examples of such averaging rules:

$$\begin{aligned}
\text{ave}(X,Y) &= \frac{1}{\text{card}\, L(X,Y)} \sum_{p \in L(X,Y)} \nu(p) \\
\text{row-ave}(X,Y) &= \frac{1}{\text{card}\, X} \sum_{p \in L(X,Y)} \nu(p) \\
\max(X,Y) &= \max_{p \in L(X,Y)} \nu(p) \\
\text{med}(X,Y) &= \operatorname*{med}_{p \in L(X,Y)} \nu(p)
\end{aligned}$$

where med is the median operation.

These notions can be naturally generalized to multiple networks $\mathcal{N} = (V, \{L_i\}, \{\nu_i\})$ where (V, L_i) are graphs and $\nu_i : L_i \to S_i$ values of lines.

5 Optimization

To express the blockmodeling problem as an optimization problem we can use the approach presented in [2, 3].

Given a set of types of connection $\mathcal{T}$ and a block $L(X,Y)$ we can determine the strongest (according to the ordering of the set $\mathcal{T}$) type T which is satisfied by $L(X,Y)$. In this case we set

$$\pi(\mu(X), \mu(Y)) = T$$

But, what to do, if no type from $\mathcal{T}$ is satisfied?

We can introduce the set of *ideal blocks* for a given type $T \in \mathcal{T}$

$$\mathcal{B}(X,Y;T) = \{B \subseteq K(X,Y) : T(B)\}$$

Table 3: Deviations measures for types of blocks

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

and define the *deviation* $\delta(X,Y;T)$ of a block $L(X,Y)$ from the nearest ideal block. In Table 2 it is shown that for types from Table 1 we can efficiently test whether the block $L(X,Y)$ is of the type T. On the basis of these characterizations we can construct also the corresponding measures of deviation (see Table 3). The quantities used in the expressions for deviations have the following meaning:

s_t	– total block sum = # of 1s in a block,
s_d	– diagonal block sum = # of 1s on a diagonal,
d	– diagonal error = $\min(s_d, n_r - s_d)$,
n_r	– # of rows in a block = $\lvert X \rvert$,
n_c	– # of columns in a block = $\lvert Y \rvert$,
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.

Note that all deviations from Table 3 are *sensitive*

$$\delta(X,Y;T) = 0 \Leftrightarrow T(L(X,Y))$$

In δ we can also incorporate values of lines ν.

Based on deviation $\delta(X,Y;T)$ we introduce the *block-error* $\varepsilon(X,Y;T)$ of $L(X,Y)$ for type T. Two examples of block-errors are

$$\varepsilon_1(X,Y;T) = w(T)\delta(X,Y;T)$$

and

$$\varepsilon_2(X, Y; T) = \frac{w(T)}{n_r n_c}(1 + \delta(X, Y; T))$$

where $w(T) > 0$ is a weight of type T.

We extend the block-error to the set of feasible types $\mathcal{T}$ by defining

$$\varepsilon(X, Y; \mathcal{T}) = \min_{T \in \mathcal{T}} \varepsilon(X, Y; T)$$

and

$$\pi(\mu(X), \mu(Y)) = \operatorname*{argmin}_{T \in \mathcal{T}} \varepsilon(X, Y; T)$$

To make π well defined we order (priorities) the set $\mathcal{T}$ and select the first type from $\mathcal{T}$ which minimizes ε.

We combine block-errors into a *total error* – blockmodeling *criterion function*

$$P(\mu; \mathcal{T}) = \sum_{(t,w) \in U \times U} \varepsilon(\mu^{-1}(t), \mu^{-1}(w); \mathcal{T})$$

For criterion function $P_1(\mu)$ we have

$$P_1(\mu) = 0 \quad \Leftrightarrow \quad \mu \text{ is an exact blockmodeling}$$

Also for P_2, we obtain an exact blockmodeling μ iff the deviations of all blocks are 0.

The obtained optimization problem can be solved by local optimization [2, 1].

Once a partitioning μ and types of connection π are determined we can compute also the values of connections. Examples of *averaging rules* for *interval* and *ordinal* networks are proposed in Table 4 where

$$\nu_t = \sum_{x \in X, y \in Y} \nu(x, y)$$

5.1 Benefits from the optimization approach to blockmodeling

In optimizational setting of the blockmodeling problem several questions can be considered:

- *ordinary blockmodeling*: Given a network $\mathcal{N}$ and set of types of connection $\mathcal{T}$, determine $\mathcal{M}$, i.e., μ, π and α;
- *evaluation of the quality of a model, comparing different models, analyzing the evolution of a network (Sampson data, [6])*: Given a network $\mathcal{N}$, a model $\mathcal{M}$, and blockmodeling μ compute the corresponding criterion function;
- *model fitting*: Given a network $\mathcal{N}$, set of types $\mathcal{T}$ and a model $\mathcal{M}$, determine μ which minimizes the criterion function.

Table 4: Averaging rules for types of blocks

	interval scale	ordinal scale
null	$\nu_t/(n_r n_c)$	0
complete	$\begin{cases} \nu_t/(n_r n_c) & \text{other} \\ \nu_t/(n_r n_c - n_r) & \text{diag}, d = 0 \end{cases}$	$\mathrm{med}\,\nu[X, Y]$
row-dominant	ν_t/n_c	$\mathrm{med}\,\nu[r, Y]$
col-dominant	ν_t/n_r	$\mathrm{med}\,\nu[X, c]$
row-regular	ν_t/n_r	$\mathrm{med}\,\nu[\max X, Y]$
col-regular	ν_t/n_c	$\mathrm{med}\,\nu[X, \max Y]$
regular	$\nu_t/\max(n_r, n_c)$	$\min(\mathrm{med}\,\nu[\max X, Y],$ $\mathrm{med}\,\nu[X, \max Y])$
row-functional	ν_t/n_r	$\mathrm{med}\,\nu[\max X, Y]$
col-functional	ν_t/n_c	$\mathrm{med}\,\nu[X, \max Y]$
density γ	$\begin{cases} \gamma\nu_t/(n_r n_c) & \text{other} \\ \gamma\nu_t/(n_r n_c - n_r) & \text{diag}, d = 0 \end{cases}$	$\mathrm{med}\,\mathrm{upper}(\gamma, \nu[X, Y])$

There are other possibilities:

- we can fit the network to a partial model and analyze the residuum 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; ...

6 Example

As an example we present some blockmodels for a Student Government Discussion network.

For a criterion function we selected $P \equiv P_1$ with all weights equal to 1. We also excluded trivial (row,col-)dominant blocks.

For each problem, determined by $P(\mu; \mathcal{T}, k)$ we performed 200 runs of local optimization. The results are presented in Table 5, where rows correspond to the number k of classes in partitions, and columns to selected types of connection $\mathcal{T}$. Entries in the table contain the corresponding minimal values of criterion function P; and as a second number, if present, the number of different optimal partitions, which are listed in Tables 6, 7 and 8.

For an example of detailed presentation we selected the solution $\mathcal{C}^a_{4,2}$. It is presented by a picture in Figure 3, and by a matrix, rearranged by classes, in Table 9. The numbers of units in the table refer to the original ordering of units in [8].

The corresponding model matrix and value matrix are given in Table 10. The model is graphically represented in Figure 4.

Table 5: Values of optimal partitions

k	str	reg	com, reg rdo, cdo	rdo, cdo	cdo	rdo	cdo(dia) reg(dia)
2	29	4	1/2	1/1	1/1	11	2
3	23	7	0/4	0/2	2/2	5	3
4	21	7	0/3	0/1	4/3	3	4
5	15	6	1/5	2/14	4/1	3	7

Table 6: Optimal partitions, $\mathcal{T} = \{$ nul, com, rdo, cdo, reg $\}$

	partition	P
$\mathcal{C}^a_{2,1}$	{m1, pm, m2, m3, m5, m6, m7, a1, a3} {m4, a2}	1
$\mathcal{C}^a_{2,2}$	{m1, a2} {pm, m2, m3, m4, m5, m6, m7, a1, a3}	1
$\mathcal{C}^a_{3,1}$	{m1, pm, m2, m3, m4, m5, m7} {m6, a3} {a1, a2}	0
$\mathcal{C}^a_{3,2}$	{m1, m2, a2} {pm, m3, m4, m5, m6, m7} {a1, a3}	0
$\mathcal{C}^a_{3,3}$	{m1, m2} {pm, a3} {m3, m4, m5, m6, m7, a1, a2}	0
$\mathcal{C}^a_{3,4}$	{m1, m4} {pm, a3} {m2, m3, m5, m6, m7, a1, a2}	0
$\mathcal{C}^a_{4,1}$	{m1, m2} {pm, m4} {m3, m5, m6, m7, a2} {a1, a3}	0
$\mathcal{C}^a_{4,2}$	{m1, m2, a2} {pm, m4} {m3, m5, m6, m7} {a1, a3}	0
$\mathcal{C}^a_{4,3}$	{m1, m2, a2} {pm, m4, m6, m7} {m3, m5} {a1, a3}	0
$\mathcal{C}^a_{5,1}$	{m1, m2} {pm, m3} {m4, a3} {m5, a1, a2} {m6, m7}	1
$\mathcal{C}^a_{5,2}$	{m1, m2, a2} {pm, m4} {m3, m5} {m6, m7} {a1, a3}	1
$\mathcal{C}^a_{5,3}$	{m1, m2, a2} {pm, m4} {m3, m6} {m5, m7} {a1, a3}	1
$\mathcal{C}^a_{5,4}$	{m1, a2} {pm, m3} {m2, a3} {m4, m5} {m6, m7, a1}	1
$\mathcal{C}^a_{5,5}$	{m1, a3} {pm, m5} {m2, m7, a1} {m3, m4} {m6, a2}	1

Table 7: Optimal partitions, $\mathcal{T} = \{$ nul, rdo, cdo $\}$

	partition	P
$\mathcal{C}^d_{2,1}$	{m1, a2} {pm, m2, m3, m4, m5, m6, m7, a1, a3}	1
$\mathcal{C}^d_{3,1}$	{m1, m4} {pm, a3} {m2, m3, m5, m6, m7, a1, a2}	0
$\mathcal{C}^d_{3,2}$	{m1, m2} {pm, a3} {m3, m4, m5, m6, m7, a1, a2}	0
$\mathcal{C}^d_{4,1}$	{m1, m2} {pm, m4} {m3, m5, m6, m7, a2} {a1, a3}	0

Table 8: Optimal partitions, $\mathcal{T} = \{$ nul, cdo $\}$

	partition	P
$\mathcal{C}^c_{2,1}$	{m1, a2} {pm, m2, m3, m4, m5, m6, m7, a1, a3}	1
$\mathcal{C}^c_{3,1}$	{m1, m4, m5} {pm, m3, m6, m7, a1, a3} {m2, a2}	2
$\mathcal{C}^c_{3,2}$	{m1, m4} {pm, m3, m5, m6, m7, a1, a3} {m2, a2}	2
$\mathcal{C}^c_{4,1}$	{m1, m2} {pm, m3, a3} {m4, m5, m6, m7, a1} {a2}	4
$\mathcal{C}^c_{4,2}$	{m1, a2} {pm, m3, a3} {m2, m4, m5} {m6, m7, a1}	4
$\mathcal{C}^c_{4,3}$	{m1, a2} {pm} {m2, m3, m4, m5, m7} {m6, a1, a3}	4
$\mathcal{C}^c_{5,1}$	{m1, a2} {pm} {m2, m4, m5, m7} {m3, a1} {m6, a3}	4

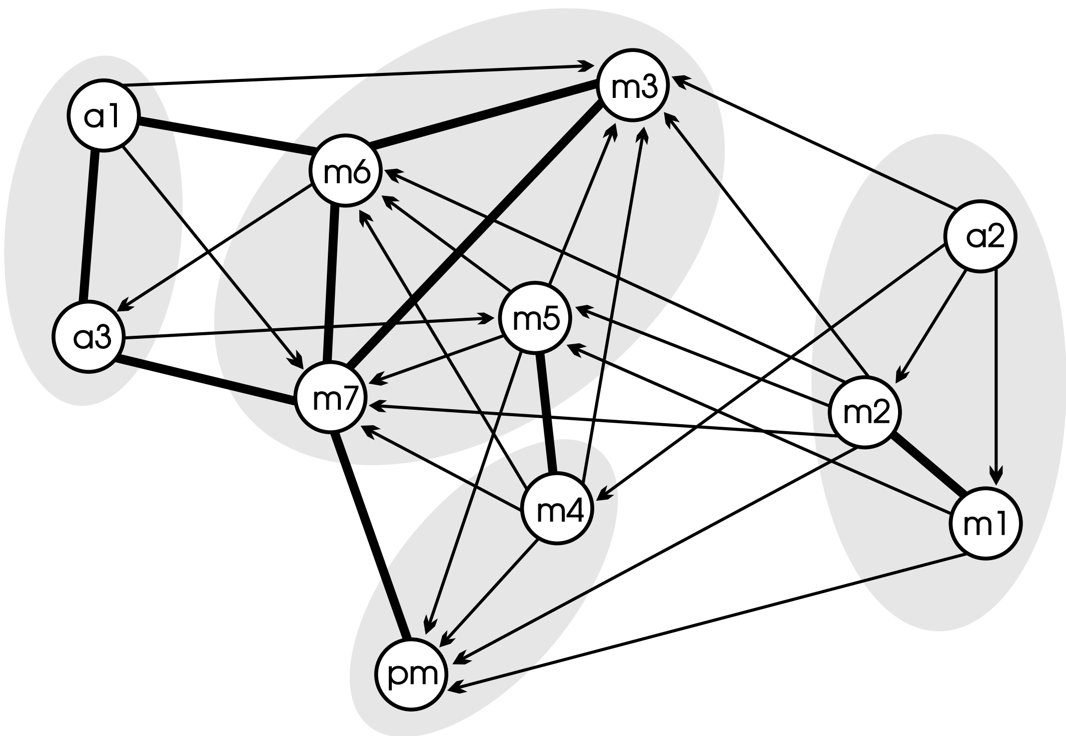

Figure 3: Partition

Table 9: Discussion Network matrix, rearranged according to $\mathcal{C}^a_{4,2}$

		m1	m2	a2	pm	m4	m3	m5	m6	m7	a1	a3
minister 1	1	0	1	0	1	0	0	1	0	0	0	0
minister 2	3	1	0	0	1	0	1	1	1	1	0	0
adviser 2	10	1	1	0	0	1	1	0	0	0	0	0
p.minister	2	0	0	0	0	0	0	0	0	1	0	0
minister 4	5	0	0	0	1	0	1	1	1	1	0	0
minister 3	4	0	0	0	0	0	0	0	1	1	0	0
minister 5	6	0	0	0	1	1	1	0	1	1	0	0
minister 6	7	0	0	0	0	0	1	0	0	1	1	1
minister 7	8	0	0	0	1	0	1	0	1	0	0	1
adviser 1	9	0	0	0	0	0	1	0	1	1	0	1
adviser 3	11	0	0	0	0	0	0	1	0	1	1	0

Table 10: Model and value matrix

	A	B	C	D	A	B	C	D
$A = \{\mathsf{m1}, \mathsf{m2}, \mathsf{a2}\}$	rdo	reg	rdo	–	1.33	1	1.5	0
$B = \{\mathsf{pm}, \mathsf{m4}\}$	–	rdo	rdo	–	0	0.5	1.25	0
$C = \{\mathsf{m3}, \mathsf{m5}, \mathsf{m6}, \mathsf{m7}\}$	–	rdo	rdo	rdo	0	1.5	2.25	1.5
$D = \{\mathsf{a1}, \mathsf{a3}\}$	–	–	cdo	com	0	0	2.5	1

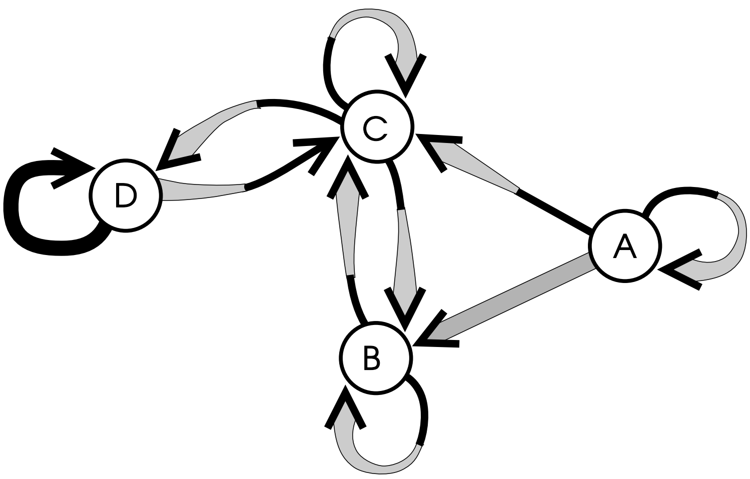

Figure 4: Model

All computations were carried out by program MODEL from a package of structure analysis programs STRAN [1]. The latest version of MODEL for PC will be available in selfextracting format by anonymous FTP from

```
uek.uni-lj.si:pub/vlado/model*.exe
```

or

```
ftp.mat.uni-lj.si:pub/datana/model*.exe
```

7 Conclusion

In the paper we proposed a generalized approach to blockmodeling of social networks. Many things have still to be elaborated:

- other types of connection and criterion functions;
- which types of connection are compatible with the hierarchy – models of models; is there some 'algebra' behind it?
- let $\mathcal{E}(\mathcal{N}, \mathcal{T})$ be the set of all equivalences compatible with $\mathcal{T}$ over $\mathcal{N}$. What can be said about the structure of this set? Can the results about the regular equivalences [4] be extended to generalized equivalences?
- assigning values also to units.

References

[1] V. Batagelj. *STRAN – STRucture ANalysis*. Manual, Ljubljana, 1991.

[2] V. Batagelj, P. Doreian, and A. Ferligoj. An optimizational approach to regular equivalence. *Social Networks*, 14:121–135, 1992.

[3] V. Batagelj, A. Ferligoj, and P. Doreian. Direct and indirect methods for structural equivalence. *Social Networks*, 14:63–90, 1992.

[4] S.P. Borgatti, M.G. Everett. The class of all regular equivalences: Algebraic structure and computation. *Social Networks*, 11:65–88, 1989.

[5] P. Doreian, V. Batagelj, and A. Ferligoj. Partitioning Networks on Generalized Concepts of Equivalence. *Journal of Mathematical Sociology*, **19**(1994)1, 1-27.

[6] P. Doreian, A. Mrvar. A Partitioning Approach to Structural Balance. Manuscript, June 1994.

[7] K. Faust. Comparison of methods for positional analysis: Structural and general equivalences. *Social Networks*, 10:313–341, 1988.

[8] V. Hlebec. Recall versus recognition: Comparison of two alternative procedures for collecting social network data. Developments in Statistics and Methodology. (A. Ferligoj and A. Kramberger, editors) Metodološki zvezki 9, FDV, Ljubljana 1993, 121-128.

[9] F. Lorrain and H.C. White. Structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1:49–80, 1971.

[10] P.E. Pattison. Network models; some comments on papers in this special issue. *Social Networks*, 10:383–411, 1988.

[11] D.R. White and K.P. Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5:193–234, 1983.

[12] R.J. Wilson, J.J. Watkins. Graphs, An Introductory Approach. John Wiley, New York, 1990.