

UNIVERSITY OF LJUBLJANA
INSTITUTE OF MATHEMATICS, PHYSICS AND MECHANICS
DEPARTMENT OF THEORETICAL COMPUTER SCIENCE
JADRANSKA 19, 1 000 LJUBLJANA, SLOVENIA

Preprint series, Vol. 40 (2002), 797

POSITIONAL ANALYSES OF SOCIOMETRIC DATA

Patrick Doreian, Vladimir Batagelj, Anuška Ferligoj

ISSN 1318-4865

Version: May 7, 2001

Math.Subj.Class.(2000): 92 H 30, 05 C 50, 05 C 70, 05 C 85,
92 G 30, 90 C 27,

Forthcoming in *Models and Methods in Social Network Analysis* edited
by P. Carrington, J. Scott and S. Wasserman, New York: Cambridge
University Press

Supported by the Ministry of Education, Science and Sport of Slovenia,
Project J1-8532.

Address: Vladimir Batagelj, University of Ljubljana, FMF, Department
of Mathematics, and IMFM Ljubljana, Department of TCS, Jadranska
ulica 19, 1 000 Ljubljana, Slovenia

e-mail: vladimir.batagelj@uni-lj.si

Ljubljana, December 28, 2001

Positional Analyses of Sociometric Data

Patrick Doreian
University of Pittsburgh

Vladimir Batagelj
University of Ljubljana

Anuška Ferligoj
University of Ljubljana

One of the major goals of social network analysis is to discern fundamental structure(s) of networks in ways that: (i) allow us to know these structures and (ii) facilitate our understanding of network phenomena. One of the most used tools for doing this is blockmodeling, a collection of methods for partitioning networks according to well specified criteria.

Initially, we use the term ‘blockmodeling’ for ‘conventional blockmodeling’ to characterize the usual approach to blockmodeling, one based on the concepts of structural equivalence (Lorrain and White, 1971) and regular equivalence (White and Reitz, 1983)¹. Our intent here is to use an optimizational approach to blockmodeling to generalize blockmodeling to consider indefinitely *many types* of blockmodels. See Batagelj *et al.* (1992a,b) for an account of optimizational methods applied to blockmodeling, Doreian *et al.*, (1994) for the extension to generalized blockmodeling, and Batagelj *et al.*, (1998) for pre-specified blockmodeling. Integral to this approach is the use of a built-in measure of the adequacy of the fit of a blockmodel.

1 Introduction

Let $\mathcal{U} = \{x_1, x_2, \dots, x_n\}$ be a finite set of *units* (that represent actors). The units are related by a binary *relation*

$$R \subseteq \mathcal{U} \times \mathcal{U}$$

which determines a *network*

$$\mathbf{N} = (\mathcal{U}, R)$$

The network can be represented by a graph with units as vertices. In general, the statement can be extended to include several relations $\{R_t\}$ and valued networks. The *location* of an actor in a network is given by the row and/or column corresponding to that actor in the

¹Included here are variants like automorphic equivalence (Faust, 1988; Pattison, 1988).

relational matrix. This extends to multiple relations. The location is the set of ties to and from all other actors in the network.

A *clustering* (partition) $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$, where $\emptyset \subset C_i \subseteq \mathcal{U}$ are *clusters*, partitions the relation R into *blocks*

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

Each cluster, C_i , is a position occupied by all of the units in the cluster, C_i . Each block is defined in terms of the units belonging to clusters C_i and C_j and all the 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.

Each partition $\mathbf{C} = \{C_i\}$ determines an equivalence relation \sim

$$u \sim v \Leftrightarrow \exists i : u, v \in C_i$$

and each equivalence relation \sim determines a partition $\mathbf{C} = \{C(u) : u \in \mathcal{U}\}$

$$C(u) = \{v \in \mathcal{U} : u \sim v\}$$

A blockmodel consists of structures obtained by identifying all units from the same cluster of the clustering \mathbf{C} and can be presented by a *reduced graph* or by a relational matrix, called an *image matrix*. The vertices in the reduced graph represent the *positions*.

Blockmodeling, as a set of empirical procedures, is based on the idea that units in a network can be grouped according to the extent to which they are equivalent, in terms of some *meaningful* definition of equivalence. In general, and without surprise, different definitions of equivalence usually lead to distinct partitions. 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.

As a simple example, we consider the Everett network (Borgatti and Everett, 1989) shown in Figure 1 where all ties are reciprocated.

1.1 Structural Equivalence

Lorrain and White (1971) provided one definition of an equivalence: Units are equivalent if they are connected to the rest of the network in *identical* ways. Such units are said to be

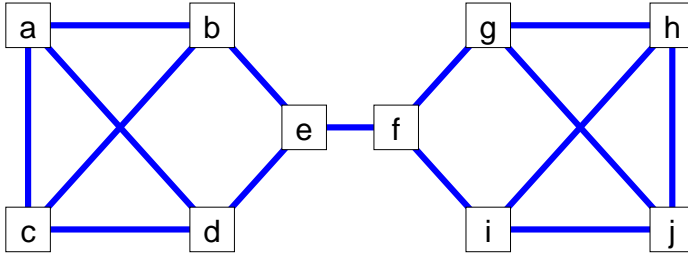


Figure 1: The Everett Network

structurally equivalent. A permutation $\varphi : \mathcal{U} \rightarrow \mathcal{U}$ is an automorphism of the relation R if and only if

$$\forall x, y \in \mathcal{U} : (xRy \Rightarrow \varphi(x)R\varphi(y))$$

The units x and y are *structurally equivalent*, we write $x \equiv y$, if and only if 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 if and only if:

- s1. $xRy \Leftrightarrow yRx$
- s2. $xRx \Leftrightarrow yRy$
- s3. $\forall z \in \mathcal{U} \setminus \{x, y\} : (xRz \Leftrightarrow yRz)$
- s4. $\forall z \in \mathcal{U} \setminus \{x, y\} : (zRx \Leftrightarrow zRy)$

On the left in Figure 1, a and c are structurally equivalent and b and d are structurally equivalent. On the right, g and i are structurally equivalent as are h and j . Note that a and c are not structurally equivalent to h and j and b and d are not structurally equivalent to h and j .

From the definition of structural equivalence it follows that only four possible ideal blocks can appear (Batagelj, Ferligoj, and Doreian, 1992b):

- Type 0. $b_{ij} = 0$
- Type 1. $b_{ij} = \delta_{ij}$
- Type 2. $b_{ij} = 1 - \delta_{ij}$
- Type 3. $b_{ij} = 1$

where δ_{ij} is the Kronecker delta function² and $i, j \in C$. The blocks of types 0 and 1 are called the *null* blocks and the blocks of types 2 and 3 the *complete* blocks.

Examples of these structural blocks are:

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

²This is defined as $\delta_{ij} = 0$ if $i \neq j$ and $\delta_{ij} = 1$ if $i = j$.

For the non-diagonal blocks $R(C_u, C_v)$, $u \neq v$, only blocks of type 0 and type 3 are admissible. An exact structural equivalence partition,

$$\mathbf{C} = \{\{a, c\}, \{h, j\}, \{b, d\}, \{g, i\}, \{e\}, \{f\}\},$$

of the Everett network is shown in Table 1. Note that the blocks in Table 1 are either null or complete. In general, specifying a set of permitted blocks defines a blockmodel *type*. In this section we are discussing the structural equivalence type of blockmodel.

Table 1: A Structural Equivalence Partition of the Everett Network

	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

1.2 Regular Equivalence

Attempts to generalize structural equivalence date back at least to Sailer (1978) and have taken various forms. Integral to all formulations is the idea that units are equivalent if they link in equivalent ways to other units that are also equivalent. Regular equivalence, as defined by White and Reitz (1983), is one such generalization.

The equivalence relation \approx on \mathcal{U} is a *regular equivalence* on network $\mathbf{N} = (\mathcal{U}, R)$ if and only if for all $x, y, z \in \mathcal{U}$, $x \approx y$ implies both

$$\text{R1. } xRz \Rightarrow \exists w \in \mathcal{U} : (yRw \wedge w \approx z)$$

$$\text{R2. } zRx \Rightarrow \exists w \in \mathcal{U} : (wRy \wedge w \approx z)$$

Another view of regular equivalence comes from using colorings. Suppose the vertices of a network are clustered, $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$, and all vertices in a cluster, C_i , are colored with the same color and all vertices are colored. The clustering \mathbf{C} determines a *coloring* $c: x \mapsto i \Leftrightarrow x \in C_i$, and vice versa. The clustering \mathbf{C} is regular if and only if

$$c(x) = c(y) \Rightarrow (c(R(x)) = c(R(y)) \wedge c(R^{-1}(x)) = c(R^{-1}(y)))$$

equivalent vertices have equally colored neighbors. See Everett and Borgatti (1996).

As was the case with structural equivalence, regular equivalence implies the existence of ideal blocks. The nature of these ideal blocks follows from the following theorem (Batagelj, Doreian, and Ferligoj, 1992a):

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

From this proposition it follows that regular equivalence produces two types of blocks:

- *null blocks* ; and
- *1-covered blocks*, which have in each row and in each column at least one 1.

We use the term ‘regular’ for these 1-covered blocks. Specifying a blockmodel in terms of only null and 1-covered blocks defines a regular equivalence type of blockmodel.

Examples of the ideal blocks for regular equivalence are:

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

An exact regular equivalence partition,

$$\mathbf{C} = \{\{a, c, h, j\}, \{b, d, g, i\}, \{e, f\}\},$$

of the Everett network is shown in Table 2. Note that the blocks are either null or regular and that this partition has the structural equivalence partition nested within it. Every structural equivalence partition is also a regular partition.

1.3 Ideal blocks and measures for blockmodels

In the empirical world, there are few exact partitions based on structural or regular equivalence ideas. In general, such blockmodels fit approximately. There is usually a non-zero number of inconsistencies when an empirical blockmodel is compared to a nearest exact ideal blockmodel. As an example consider the network of the Little League team shown in Figure 2. In this figure and all other empirical networks considered in this chapter, thick lines represent reciprocated ties between pairs of actors. Thin lines with arrows represent

Table 2: A Regular Equivalence Partition of the Everett Network

	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

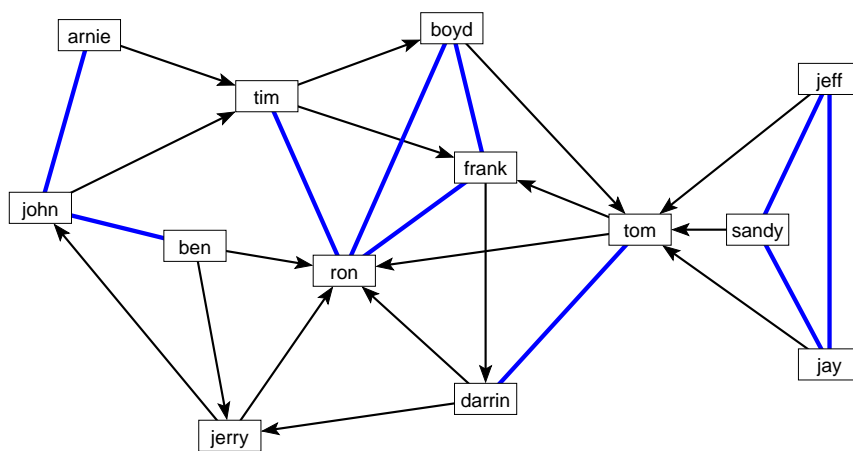


Figure 2: A Little League Baseball Team Network

unreciprocated ties. The only actors that are structurally equivalent are Jay, Jeff, and Sandy, and any clustering together of other actors will yield a blockmodel with inconsistencies when compared to an ideal structural equivalence blockmodel.

In broad terms, blockmodeling tools are empirical partitioning (clustering) procedures. Usually, there are few formal assessments of how well the blockmodels fit the data. In the approach taken here, we present ways of blockmodeling that incorporate *measures of adequacy*. The adequacy of these procedures is pursued as follows.

Assume, as before, that we have a single relation network $\mathbf{N} = (\mathcal{U}, R)$. Let Θ denote

the set of all equivalence relations of a selected type (for example, regular or structural equivalence) over \mathbf{N} . Every equivalence relation $\sim \in \Theta$ on \mathcal{U} determines a partition \mathbf{C} of \mathcal{U} .

Let Φ denote the set of all partitions into k clusters, called also the set of *feasible* clusterings. If we are able to construct a criterion function $P(\mathbf{C})$, $\mathbf{C} \in \Phi$ with the properties:

$$\text{P1. } P(\mathbf{C}) \geq 0 \qquad \text{P2. } P(\mathbf{C}) = 0 \Leftrightarrow \sim \in \Theta,$$

then we can express the problem of establishing a partition of a network, in terms of a specific type of equivalence, as a clustering problem where the task is to determine the clustering $\mathbf{C}^* \in \Phi$ for which $P(\mathbf{C}^*) = \min_{\mathbf{C} \in \Phi} P(\mathbf{C})$. If there are exact equivalences in Φ , then (by P2) the minimal value of $P(\mathbf{C})$ is 0. In the case where there is no exact equivalence, then $\forall \mathbf{C} \in \Phi$, $P(\mathbf{C}) > 0$, the optimization approach gives the solutions which differ the least from some ideal case. There can be multiple equally well fitting partitions for a well defined blockmodel type.

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 (i.e. ideal blocks) within each cluster and between clusters according to the specific type of 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 *local* error – the difference (inconsistency) between the block $R(C_u, C_v)$ and the corresponding ideal block B . The function d has to be compatible (Batagelj *et al.*, 1992b) with the selected type of equivalence.

1.4 A Clustering Algorithm

In the case of the direct clustering approach, where an appropriate criterion function to capture the selected equivalence is constructed, local optimization clustering procedure (a relocation algorithm) can be used to solve the given blockmodeling problem (Batagelj *et al.*, 1992b):

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

Usually, the *neighborhood* is determined by two transformations: *moving* a unit from one cluster to another cluster; and *interchanging* of two units from different clusters.

To obtain a ‘good’ solution and an impression of its quality we repeat the procedure with different (random) initial partitions \mathbf{C} . If the procedure is repeated many times (i.e. some hundreds of times) all or most of partitions of selected type of equivalence (structural or regular) in a given network can be found.

1.5 A Criterion for Structural Equivalence

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

$$d(R(C_u, C_v), B) = \sum_{x \in C_u, y \in C_v} |r_{xy} - b_{xy}|.$$

In this expression, 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 null blocks. These two types of inconsistencies can be weighted differently. It is easy to verify that a criterion function $P(\mathbf{C})$ defined in this fashion is *sensitive* (Batagelj *et al.*, 1992b) to structural equivalence:

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

The unique best fitting partition of the Little League network (for four positions based on structural equivalence) is shown in Table 3. Note that the four positions define the 16 blocks in the blockmodel. The pattern of nearest ideal block types is at the bottom of Table 3.

One product of the direct partitioning approach is a count of the inconsistencies, by block, of the partition. Corresponding to Table 3, the distribution of inconsistencies is:

2	1	1	0
2	0	1	0
5	1	7	0
0	0	0	0

There are eight blocks with zero inconsistencies. The 20 inconsistencies are distributed across the other eight blocks. For C_1 , there are 2 inconsistencies in its block. These are no ties from Frank and Boyd to Tim in what should be a block of type 4. For the link from C_1 to C_2 , there is one inconsistency. Boyd sends a tie to Tom in what, otherwise, would be a null block. Similarly, the lone link from Frank in C_1 to Darrin in C_3 is an inconsistency. For the ties from Tom (in C_2) to boys in C_1 , either the two 0’s or the two 1’s can be viewed as contributing two inconsistencies to the total count of inconsistencies. When we treat the corresponding ideal block as null, the ties from Tom to Ron and Frank are the inconsistencies. The tie from Tom to Darrin (in cluster C_3) is another inconsistency.

Table 3: Little League Structural Equivalence Partition

		1	3	4	5	2	6	10	11	12	13	7	8	9
C_1	Ron	1	0	1	1	1	0	0	0	0	0	0	0	0
	Frank	3	1	0	1	0	0	0	0	1	0	0	0	0
	Boyd	4	1	1	0	0	1	0	0	0	0	0	0	0
	Tim	5	1	1	1	0	0	0	0	0	0	0	0	0
C_2	Tom	2	1	1	0	0	0	0	0	1	0	0	0	0
C_3	John	6	0	0	0	1	0	0	0	0	1	1	0	0
	Jerry	10	1	0	0	0	0	1	0	0	0	0	0	0
	Darrin	11	1	0	0	0	1	0	1	0	0	0	0	0
	Ben	12	1	0	0	0	0	1	1	0	0	0	0	0
	Arnie	13	0	0	0	1	0	1	0	0	0	0	0	0
C_4	Jeff	7	0	0	0	0	1	0	0	0	0	0	0	1
	Jay	8	0	0	0	0	1	0	0	0	0	0	1	0
	Sandy	9	0	0	0	0	1	0	0	0	0	0	1	1










complete	null	null	null
null	null	null	null
null	null	null	null
null	complete	null	complete

There are 5 ties from boys in C_3 to boys C_1 and they are all inconsistent with a null block. In a similar fashion, the 7 ties among the boys in cluster C_3 are all inconsistencies. There are no inconsistencies among the ties from boys in cluster C_4 . For now, we note that 12 of inconsistencies are concentrated in two of the cells. It seems reasonable that, for the model as a whole, the inconsistencies be distributed across all of the cells in roughly the same fashion (controlling for block size). Having the inconsistencies piling up in a small number of cells suggests that we take a closer look at the partition and the distribution of inconsistencies.

2 The Generalized Blockmodeling Approach

The logic and history of blockmodeling took the form of defining types of equivalences and then searching for partitions that were thought to be consistent with the specified equivalences. Under the, so called, indirect approach (for example in Burt, 1976; Breiger *et al.*, 1975) a relational matrix is turned into a matrix of (dis)similarities and clustered. The di-

Table 4: Block Types

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

rect approach to blockmodeling (Batagelj *et al.*, 1992 a,b) takes advantage of the result that structural and regular equivalence each implied a (small) set of permitted blocks in an ideal blockmodel image. In this way, it was straightforward to specify a criterion function that captured the difference between an empirical blockmodel and a (or the) nearest ideal blockmodel. Doreian *et al.* (1994) present evidence that shows that blockmodeling partitions established with the direct approach are usually better - and are never worse - than those established with indirect methods.

The logic of generalized blockmodeling is to *start with sets of permitted ideal blocks*. An appropriate generalization of the equivalence idea is one where each block, of a particular partition, is free to conform to a different type. This led Batagelj (1997) and Doreian *et al.* (1994) to the definition of several types of connection inside and between the clusters as different types of ideal blocks. Some of them are characterized in Table 4. Table 5 shows these block types in more detail. The evaluations of inconsistencies, functions $d(R(C_u, C_v), B; T)$, are defined specifically for each type T of ideal blocks.

2.1 Revisiting the Little League Network

With this set of block types in mind, reconsider the partition shown in Table 3. We could recode the nearest ideal block types as:

Table 5: Examples of Blocks with Types of Connections

C_i	<table style="border-collapse: collapse; width: 100%;"> <tr><th style="border: none;"></th><th style="border: none;">C_j</th></tr> <tr><td style="border: none;">1</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">1</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">1</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">1</td><td style="border: none;">1</td></tr> </table>		C_j	1	1	1	1	1	1	1	1	C_i	<table style="border-collapse: collapse; width: 100%;"> <tr><th style="border: none;"></th><th style="border: none;">C_j</th></tr> <tr><td style="border: none;">0</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">1</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> </table>		C_j	0	1	1	1	0	0	0	0	C_i	<table style="border-collapse: collapse; width: 100%;"> <tr><th style="border: none;"></th><th style="border: none;">C_j</th></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">1</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> </table>		C_j	0	0	0	0	1	1	0	0
	C_j																																		
1	1																																		
1	1																																		
1	1																																		
1	1																																		
	C_j																																		
0	1																																		
1	1																																		
0	0																																		
0	0																																		
	C_j																																		
0	0																																		
0	0																																		
1	1																																		
0	0																																		
	complete		row-dominant		col-dominant																														
C_i	<table style="border-collapse: collapse; width: 100%;"> <tr><th style="border: none;"></th><th style="border: none;">C_j</th></tr> <tr><td style="border: none;">0</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">1</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">1</td><td style="border: none;">1</td></tr> </table>		C_j	0	1	1	0	0	0	1	1	C_i	<table style="border-collapse: collapse; width: 100%;"> <tr><th style="border: none;"></th><th style="border: none;">C_j</th></tr> <tr><td style="border: none;">0</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">1</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">1</td></tr> </table>		C_j	0	1	0	1	1	0	0	1	C_i	<table style="border-collapse: collapse; width: 100%;"> <tr><th style="border: none;"></th><th style="border: none;">C_j</th></tr> <tr><td style="border: none;">0</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">1</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">1</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> </table>		C_j	0	1	1	0	1	1	0	0
	C_j																																		
0	1																																		
1	0																																		
0	0																																		
1	1																																		
	C_j																																		
0	1																																		
0	1																																		
1	0																																		
0	1																																		
	C_j																																		
0	1																																		
1	0																																		
1	1																																		
0	0																																		
	regular		row-regular		col-regular																														
C_i	<table style="border-collapse: collapse; width: 100%;"> <tr><th style="border: none;"></th><th style="border: none;">C_j</th></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> </table>		C_j	0	0	0	0	0	0	0	0	C_i	<table style="border-collapse: collapse; width: 100%;"> <tr><th style="border: none;"></th><th style="border: none;">C_j</th></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">1</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> </table>		C_j	0	0	0	0	1	0	0	0	C_i	<table style="border-collapse: collapse; width: 100%;"> <tr><th style="border: none;"></th><th style="border: none;">C_j</th></tr> <tr><td style="border: none;">1</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">1</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> <tr><td style="border: none;">0</td><td style="border: none;">0</td></tr> </table>		C_j	1	0	0	1	0	0	0	0
	C_j																																		
0	0																																		
0	0																																		
0	0																																		
0	0																																		
	C_j																																		
0	0																																		
0	0																																		
1	0																																		
0	0																																		
	C_j																																		
1	0																																		
0	1																																		
0	0																																		
0	0																																		
	null		row-functional		col-functional																														

reg	col-regular	null	null
row-regular	null	row-regular	null
row-regular	col-regular	row-reg	null
null	complete	null	complete

and the pattern of inconsistencies becomes:

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

Of course, we do *not* advocate that blockmodels obtained under one set of permitted block types then be reinterpreted with a different set of block types. We use this example as a simple way of thinking about alternative block types. Had we specified a set of permitted block types as (complete, regular, row-regular, column-regular, and null) and then fitted the (generalized) blockmodel, this partition would be reached. As another example, consider the set of permitted block types as (complete, regular, row-dominant, column-dominant, and null). This specification leads to the generalized blockmodel in Table 6. There are no inconsistencies with the specified blockmodel.

Table 6: A Generalized Partition of the Little League Network

		1	2	5	6	10	11	12	13	7	8	9	3	4
C_1	Ron	1	0	0	1	0	0	0	0	0	0	0	1	1
	Tom	2	1	0	0	0	0	1	0	0	0	0	1	0
	Tim	5	1	0	0	0	0	0	0	0	0	0	1	1
	John	6	0	0	1	0	0	0	1	1	0	0	0	0
	Jerry	10	1	0	0	1	0	0	0	0	0	0	0	0
	Darrin	11	1	1	0	0	1	0	0	0	0	0	0	0
C_2	Ben	12	1	0	0	1	1	0	0	0	0	0	0	0
	Arnie	13	0	0	1	1	0	0	0	0	0	0	0	0
C_3	Jeff	7	0	1	0	0	0	0	0	0	0	1	1	0
	Jay	8	0	1	0	0	0	0	0	0	0	1	0	1
	Sandy	9	0	1	0	0	0	0	0	0	0	1	1	0
C_4	Frank	3	1	0	0	0	0	1	0	0	0	0	0	1
	Boyd	4	1	1	0	0	0	0	0	0	0	0	1	0

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

Pictorially, we have the partitioned network as shown on the left side of Figure 3 and the image network as shown on the right side of Figure 3. This image provides an alternative (generalized) blockmodel with a different interpretation compared to the blockmodel image in Table 3. It reveals a very clear core-periphery structure defined in terms of row dominance and column dominance. The core of the team is found in cluster C_1 whose diagonal block is regular. The row dominant link from C_1 to C_2 comes from John's ties to Arnie and Ben and the column dominant link from C_2 to C_1 comes from the ties from Arnie and Ben (in C_2)

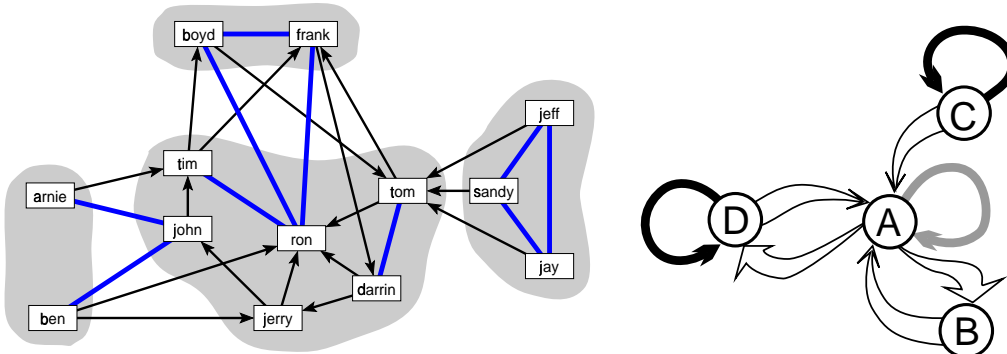


Figure 3: Generalized Partition of the LL Network and its Image Graph

to John (in C_1). The tie from C_1 to C_3 is null as there are no ties from boys in C_1 to boys in C_3 . The col-dominant tie from C_3 to C_1 comes from all the boys in C_3 choosing Tom in C_1 . The row dominant link from C_1 to C_4 comes from Ron choosing both Frank and Boyd (as does Tim). The column dominant link to C_1 from C_4 comes from Frank and Boyd in C_4 both choosing Ron in C_1 . Finally, both C_3 and C_4 are complete (diagonal) blocks.

We note an additional feature of generalized blockmodeling: a preference for the order of permitted block types. In generating Table 6, we gave preference to row-dominance and column dominance as block types (compared to row-regular and column-regular block types). It is possible to view the ties from C_1 to both C_2 and C_4 as row-regular. Similarly, all of the ties from C_2 , C_3 and C_4 to C_1 can be viewed as row regular. Were preference given to those block types and *if* the same partition was returned, we could interpret these blocks as row and column regular blocks. However the ‘if’ in the previous sentence should be noted - the generalized blockmodel as a whole is specified *prior* to an analysis given that specification rather be one that is reinterpreted after some other analysis has be completed.

3 Pre-specified Blockmodels

In revisiting the Little League baseball team network, we began doing more than just specifying block types. In addition to specifying types, it is possible to require also that particular block types go in particular places in the blockmodel. When a (generalized or conventional) blockmodel has this additional specification, we call it a ‘pre-specified blockmodel’. Our first illustration of this takes the form of a baboon grooming network.

Table 7: Permuted Grooming Ties for a Two Cluster Model

		1	3	4	6	8	10	11	2	5	7	9	12
f_1	1	.	.	1	.	.	.	1	1	1	.	.	1
f_2	3	.	.	1	1	.	1	1	.	1	1	.	1
f_3	4	1	1	.	1	.	1	.	.	1	1	.	1
f_4	6	.	1	1	.	.	1	.	.	1	.	.	1
f_5	8	1	.	.	1	1	1	.
f_6	10	.	1	1	1	1
f_7	11	1	1
m_1	2	1
m_2	5	1	1	1	1	1
m_3	7	.	1	1	.	1
m_4	9	1
m_5	12	1	1	1	1

regular	column-regular
row-regular	null

3.1 A Baboon Network as A Core-Periphery Structure

Table 7 shows a two-cluster (four blocks) partition of a baboon grooming network. A first examination of these ties provided a strong clue as to how to partition this network: males do not groom other males. This suggests that the grooming structure is centered on the females and this observation is a first step in constructing a blockmodel as a ‘core-periphery’ structure. We retain the idea of a core as a block where ties are sufficiently dense. One operationalization of this could be that diagonal blocks are complete. This may be too stringent if a coherent core does not require that *all* actors be mutually linked. However, for social cohesion in the core, *enough* pairs need to be in mutual ties. Another possible specification is that the core block is regular – in the sense of regular equivalence – so that each member in the core is (mutually) linked to at least one other member of the core. The baboons (males) in the periphery do not groom each other so their diagonal block is null. The males are linked to females in the core in such a way as each male is groomed by at least one female. Table 7 shows the pre-specified model implied by this argument below the empirically fitted blockmodel. The distribution of ties in Table 7 conforms exactly to the pattern of the pre-specified blockmodel. In this sense, the pre-specified blockmodel has been ‘tested’ and supported as an empirical hypothesis.

It is possible to fit a finer grained partition (with more positions). Some of the females

Table 8: A Four Position Model Fitted to Baboon Network

	3	4	10	1	6	8	11	5	7	12	2	9	
f_2	3	.	1	1	.	1	.	1	1	1	.	.	
f_3	4	1	.	1	1	1	.	.	1	1	1	.	.
f_6	10	1	1	.	.	1	1	
f_1	1	.	1	1	1	.	1	.	
f_4	6	1	1	1	1	.	1	.	
f_5	8	.	.	1	1	1	.	1	
f_7	11	1	.	.	1	
m_2	5	1	1	.	1	1	1	
m_3	7	1	1	.	.	.	1	
m_5	12	1	1	.	1	1	
m_1	2	.	.	.	1	
m_4	9	1	

	C_1	C_2	C_3	C_4
C_1	regular	col-regular	col-regular	null
C_2	row-regular	null	col-regular	col-regular
C_3	row-regular	row-regular	null	null
C_4	null	row-regular	null	null

do not groom males and a finer grained pre-specified model takes the following form:

regular	col-regular or null	col-regular	col-regular
row-regular or null	regular or null	col-regular	col-regular
row-regular or null	row-regular or null	null	null
row-regular or null	row-regular or null	null	null

With this specification in mind, fitting a generalized blockmodel resulted in the partition shown in Table 8.

The fitted pre-specified blockmodel has two inconsistencies in the cell corresponding to the diagonal block for C_2 . In this baboon troop, the blocks corresponding to ties between C_1 and C_2 happen to be regular - but we think that this may be too strong as a general specification of core-periphery models. A more extended discussion of core-periphery models is provided in Doreian *et al.* (2001).

Table 9: A Social Relation as A Ranked Clusters System

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>m</i>	<i>n</i>	<i>o</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>p</i>	<i>q</i>	<i>r</i>	<i>s</i>
<i>a</i>	.	1	1	1
<i>b</i>	1	.	1
<i>c</i>	1	1
<i>d</i>	.	1	.	.	1
<i>e</i>	.	.	1	1
<i>j</i>	.	.	.	1	.	.	<i>I</i>	1
<i>k</i>	1	.	.	1
<i>l</i>	1	1	1
<i>m</i>	.	.	1	1	1	1
<i>n</i>	.	.	1	.	1	.	.	.	1
<i>o</i>	1	<i>I</i>
<i>f</i>	1	1	1	1
<i>g</i>	1	1	.	1	1
<i>h</i>	1	1	1	.	1
<i>i</i>	1	1	1	1	1	.	.	1	.	.
<i>p</i>	.	.	.	1	1	1	.	.	.	1	.	.	1	.	.
<i>q</i>	.	.	.	1	1	1	1	.	1	1
<i>r</i>	.	.	.	1	1	1	<i>I</i>	1	.	1
<i>s</i>	.	.	.	1	1	1	.	.	.	1	1	.

3.2 Ranked Clusters Models of Stratified Sociometric Systems

A classical result from Harary, Norman, and Cartwright (1965) says that any directed graph factored by its strong connectivity relation gives an acyclic structure (model).

A line of research was started by Davis and Leinhardt (1972) when they formulated a ‘ranked-clusters’ model for stratified sociometric systems. In their formulation, there are distinct patterns in the location of mutual, asymmetric and null ties in a ranked clusters model. The mutual ties are only in cliques (as maximal complete subgraphs). These cliques are distributed across ranks so that asymmetric ties always go in one direction (usually up) and they never occur in the opposite direction. Null ties can be anywhere except within cliques. Put differently, in the pre-specified blockmodel, diagonal blocks are specified as complete, blocks above the diagonal are specified as null and only blocks below the diagonal can be non-null.

Doreian *et al.* (2000) proposed a slightly more general blockmodel for ranked clusters systems. All that differs in their version is the specification of *symmetric* diagonal blocks

as a way of weakening the requirement of full cliques in the diagonal blocks. Formally, the new symmetric block is specified as follows. A (diagonal³) block is symmetric if

$$\forall x, y \in C_i \times C_j : (xRy \Leftrightarrow yRx).$$

The *new blockmodel type* specifies symmetric blocks on the diagonal and null blocks above the diagonal. The bulk of their paper was devoted to symmetric-acyclic decompositions of networks with generalized blockmodels in a complementary (but secondary) role. The acyclic requirement is captured by having null blocks above the diagonal. The ranked-clusters blockmodel is defined by a distinctive *pattern* of the location of block types, a well defined pre-specified blockmodel.

A hypothetical ranked clusters blockmodel is given in Table 9 and blocked in a way to show the ranked clusters structure. The italicized elements in the diagonal blocks are ties that are inconsistent with the symmetric diagonal block requirement. The bolded ties above the diagonal are inconsistencies with the acyclic requirement that there be no cycles linking non-diagonal blocks.

One departure with regard to blockmodeling is illustrated with the ranked-clusters model as used here. We can view the two types of inconsistencies as differentially important and weight them accordingly. Inconsistencies with the strict ranking (acyclic) requirement seem more consequential and are weighted more heavily than inconsistencies with the symmetry requirement⁴.

3.2.1 A Trust Relation in an Organization

Figure 4 shows an exact ranked-clusters model taken from Doreian (2001a). The data come from a study by French (1963) of salesmen in a competitive working environment where trust was of great importance. The relation depicted is trust ties for the first of three time points. The heavy lines depict the symmetric (mutually trusting) ties and the thin lines represent unreciprocated trust ties. The actors linked only by symmetric ties go into a set of six positions whose (diagonal) blocks are symmetric. These are shaded in Figure 4. The overall structure can be described roughly as one with parallel ranked-clusters subsystems. One has the set {Blumberg, Beere, Plotkin} as its top while the other has {Holzer, Murphy, Brim} at the top.

3.2.2 A Children's Network

Figure 5 shows a network for a group of girls in a sixth grade children's network where the data come from Jennings (1948). (See Doreian (2001b) for other examples of ranked-

³While this can be extended to include non-diagonal blocks, this extension is not relevant here.

⁴We are experimenting with different weighting regimes but, in the main, have weighted the inconsistencies with the acyclic requirement at 10 times the weight for the inconsistencies with symmetry. In essence, we are solving a 2-criteria clustering problem by reducing it to a single criterion problem.

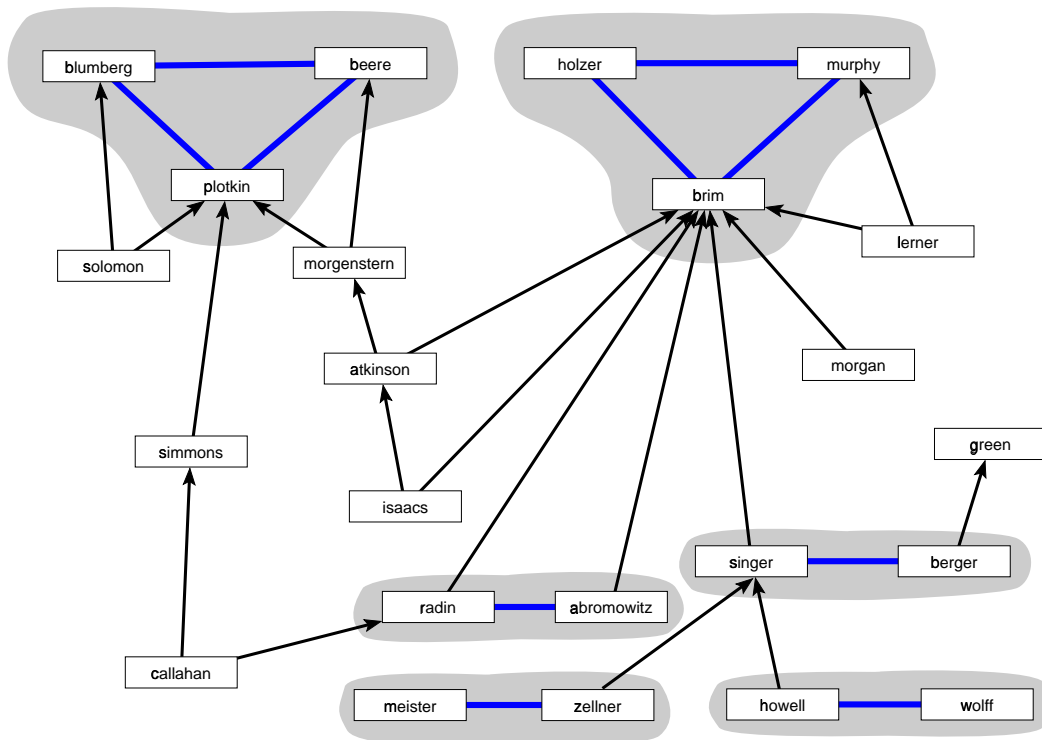


Figure 4: An Exact Ranked-Clusters Model of a Trust Relation

clusters models of children's networks.) While there are boys in the classroom there are hardly any ties going between children of different genders. So, for this classroom, focusing on just the girls does not distort their network. As for the trust example, the thick lines depict symmetric ties with the thin lines representing unreciprocated ties. There are two inconsistencies with a ranked-clusters model. One is the unreciprocated link from g_{14} to g_{12} within a diagonal block while the other is the link from g_{12} in a high ranking cluster to g_{16} in a low ranking cluster. Figure 5 show the ranked-clusters model where: $G_1 = \{g_{12}, g_{13}, g_{14}, g_{18}\}$; $G_2 = \{g_{19}, g_{20}, g_{21}\}$; $G_3 = \{g_6, g_8\}$; $G_4 = \{g_3, g_{15}, g_{16}, g_{17}\}$ and $G_5 = \{g_1, g_2\}$. These clusters having more than one member are represented as circles in the image graph on the right side of Figure 5. To keep the generalized blockmodel image simple, the self loops for these positions have not been drawn. The remaining 'clusters' are singletons and have represented as squares in Figure 5.

We suggest that when efforts are made to characterize ranked or hierarchical sociometric

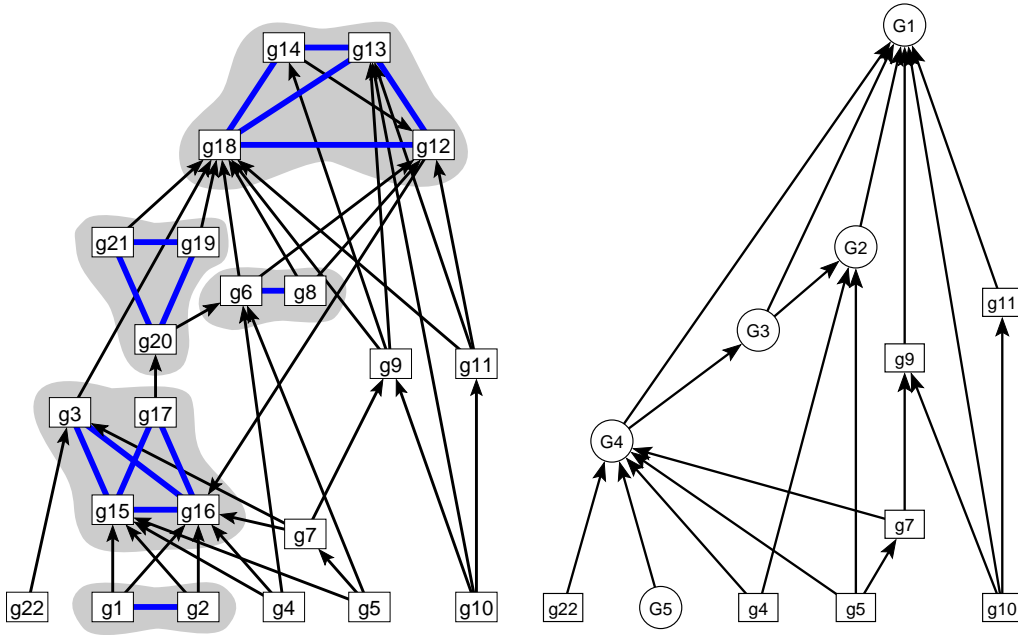


Figure 5: A Network of Ties for Girls in a Sixth Grade Classroom and its Ranked-Clusters Model

systems that the use of ranked-clusters models (when they fit) as one way of characterizing the hierarchical structure. The pre-specified blockmodeling is supported by the program MODEL 2 (Batagelj, 1997).

4 Formalization of Blockmodeling

We finish by providing a formal statement of generalized blockmodeling that is applicable to all types of blockmodels. The point of departure is, as before, a network, \mathbf{N} , with a set of units, \mathcal{U} , and a relation $R \subseteq \mathcal{U} \times \mathcal{U}$. Let \mathcal{Z} be a set of positions or images of clusters of units. Let $\mu : \mathcal{U} \rightarrow \mathcal{Z}$ denote a mapping which maps each unit to its position. The cluster of units $C(t)$ with the same position $t \in \mathcal{Z}$ is

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

Therefore

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

is a partition (clustering) of the set of units \mathcal{U} . This is shown in Figure 6 where the clusters C_i and C_j are mapped under μ to their positions in the image. The general problem is to determine the nature of the tie between the positions i and j (to which C_i and C_j are mapped in the image). To do this we have to determine the type, T of tie (reflecting the structure of the block $R(C_i, C_j)$) and the value a of the tie which summarizes the values $a_{x,y}$.

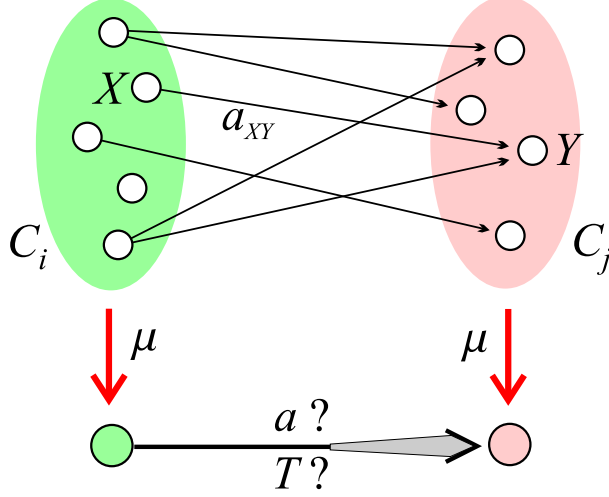


Figure 6: Generalized Blockmodeling Scheme for Image Types

A (generalized) blockmodel is an ordered quintuple $\mathbf{M} = (\mathcal{Z}, K, \mathcal{T}, \pi, \alpha)$ where:

- \mathcal{Z} is a set of positions;
- $K \subseteq \mathcal{Z} \times \mathcal{Z}$ is a set of *connections* between positions;
- \mathcal{T} is a set of predicates used to describe the types of connections between clusters 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.
- A mapping $\alpha : K \rightarrow Q$ where Q is a set of ‘averaging rules’.

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

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

and

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

Let \sim be an equivalence relation over \mathcal{U} . It partitions the set of units \mathcal{U} into clusters

$$[x] = \{y \in \mathcal{U} : x \sim y\}$$

We say that \sim is *compatible* with \mathcal{T} or is a \mathcal{T} -equivalence over a network \mathbf{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 $\mathcal{T} = \{\text{nul}, \text{reg}\}$ reduces to the usual definition of regular equivalence. Similarly, compatibility for $\mathcal{T} = \{\text{nul}, \text{com}\}$ reduces to structural equivalence.

For a compatible equivalence \sim the mapping $\mu: x \mapsto [x]$ determines a blockmodel with $\mathcal{Z} = \mathcal{U}/\sim$.

4.0.3 Criterion Functions

One of the possible ways of constructing a criterion function that directly reflects the selected type of equivalence is to measure the departure of a clustering from an ideal one with perfect relations within each cluster and between clusters according to the selected type of equivalence. Notationally, this is the same as described in Section 1.3. For details, see Batagelj (1997). For generalized blockmodels we add the following specification: Given a set of types of connection \mathcal{T} and a block $R(C_u, C_v)$, $C_u, C_v \subseteq \mathcal{U}$, we can determine the strongest (according to the ordering of the set \mathcal{T}) type T which is satisfied the most by $R(C_u, C_v)$. In this case we set

$$\pi((\mu(C_u), \mu(C_v))) = T$$

Given the specification of a generalized blockmodel and the specification of a criterion function, as written in Section 1.3, but with the richer set of connection types, the block-modeling problem is solved using a local optimization procedure (relocation algorithm) described in Section 1.4. For a general approach to clustering relational and multivariate data, see Batagelj and Ferligoj (2000).

5 Conclusion

By starting with new block types (and hence new blockmodel types) and allowing each block to have its own characterization means that the number of blockmodel types can be increased indefinitely. New block types can be defined formally (as done in Table 5) or can have substantive foundations (as is the case with the ranked-clusters model). Structural balance theory provides another example of a substantively based (generalized) blockmodel where an ideal blockmodel takes the form of having diagonal blocks containing only positive and null ties and off-diagonal blocks having only negative and null ties. See Doreian

and Mrvar (1996). The way is clear for using a much richer set of blocks and blockmodel types and the construction of richer social theories.

There are two broad caveats to this sweeping claim. First, the choice of ideal block types and blockmodel types *must be specified on substantive grounds prior to an analysis*. This specification can take a weak form where only the block types are specified or a strong form where *both* the block types and their locations in a generalized blockmodel are specified. In this context, we use the term ‘pre-specified’ (generalized) blockmodel to emphasize that in this strong specification, more is involved than selecting block types. The weak specification corresponds to an *inductive* (exploratory) use of generalized blockmodeling while strong specification corresponds to a *deductive* (confirmatory) use of generalized blockmodeling where the blockmodel is pre-specified and *fitted*. In this sense, the pre-specified blockmodel is viewed as an hypothesis and is tested. If there are too many inconsistencies when this blockmodel is fitted, the hypothesis is not supported in the relational data. Second, with a rich array of block types available, it is possible always to locate/fit a generalized blockmodel that fits exactly (with zero inconsistencies). When generalized blockmodels are established ‘blindly’ (i.e. with many block type switched on) it seems that this is an analogue to ‘capitalizing on chance’ in statistically fitting empirical equations. We doubt that generalized blockmodels established blindly have any substantive or practical value. We acknowledge that do not have a ‘theory of errors’ in fitting generalized blockmodels at this time. As a result, the boundary between generalized blockmodels that fit and those that do not fit is fuzzy. However, we argue that placing substance first goes a long way in protecting us against the risk of fitting nonsense generalized blockmodels. Establishing a procedure for fitting generalized blockmodels with a well founded theory of errors is a future task. For now our attention is on the substantive gains that become available with an expanded set of block and blockmodel types.

In general, for a given network, a set of ideal blocks is selected, a reduced model is formulated, and partitions are established by solving a clustering problem by minimizing the criterion function. The use of optimization procedures is crucial and provides considerable benefits:

- In conventional blockmodeling in inductive mode, it permits the establishment of empirical blockmodels;
- Given a specific type of blockmodel, and through time data, the value of the criterion function can be tracked to measure the quality of the fit of the model through time. Structural balance theory provides an obvious example.;
- When pre-specified models are used in deductive mode, optimization methods permit the testing of these models as discussed above. The pre-specified blockmodeling starts with a blockmodel specified, in terms of substance, *prior to an analysis*.

References

- [1] BATAGELJ, V. (1997), "Notes on Blockmodeling," *Social Networks*, 19, 143–155. presented at 3rd European Social Networks Conference, München, 1993.
- [2] BATAGELJ, V., DOREIAN, P., and FERLIGOJ, A. (1992a), "An Optimizational Approach to Regular Equivalence," *Social Networks*, 14, 121–135.
- [3] BATAGELJ, V. and FERLIGOJ, A. (2000), "Clustering Relational Data," in *Data Analysis: Scientific Modeling and Practical Application*, (ed.: W. Gaul, O. Opitz, M. Schader), Studies in classification, data analysis, and knowledge organization. Berlin: Springer, pp. 3-15.
- [4] BATAGELJ, V., FERLIGOJ, A., and DOREIAN, P. (1992b), "Direct and Indirect Methods for Structural Equivalence," *Social Networks*, 14, 63–90.
- [5] BATAGELJ, V., FERLIGOJ, A., and DOREIAN, P. (1998), "Fitting Pre-specified Blockmodels," in *Data Science, Classification, and Related Methods*, (C. Hayashi, N. Oshumi, K. Yajima, H.-H. Bock, Y. Baba eds.), Tokyo: Springer-Verlag, pp. 199-206.
- [6] BORGATTI, S. P., and EVERETT, M. G. (1989), "The Class of all Regular Equivalences: Algebraic Structure and Computation," *Social Networks*, 11, 65–88.
- [7] BORGATTI, S. P., and EVERETT, M. G. (1992), "Notions of Positions in Social Network Analysis," in *Sociological Methodology*, Ed., P. V. Marsden, San Francisco: Jossey-bass, pp. 1–35.
- [8] BREIGER, R. L., BOORMAN, S. A., and ARABIE, P. (1975), "An Algorithm for Clustering Relational Data with Applications to Social Network Analysis and Comparison to Multidimensional Scaling," *Journal of Mathematical Psychology*, 12, 328–383.
- [9] BURT, R. S. (1976), "Positions in Networks," *Social Forces*, 55, 93–122.
- [10] DAVIS, J. A., and LEINHARDT, S. (1972), "The Structure of Positive Interpersonal Relations in Small Groups," in *Sociological Theories in Progress, Volume 2*, Ed., J. Berger, Boston: Houghton Mifflin, pp. 218–251.
- [11] DOREIAN, P., (2001a), "Ranked Clusters Models of Stratified Sociometric Systems" *Department of Sociology*, University of Pittsburgh.
- [12] DOREIAN, P., (2001b), "Social Environments and Network structures" *Department of Sociology*, University of Pittsburgh.

-
- [13] DOREIAN, P., BATAGELJ, V., and FERLIGOJ, A. (2000), "Symmetric-Acyclic Decompositions of Networks" *Journal of Classification*, 17, 3-28.
- [14] DOREIAN, P., BATAGELJ, V., and FERLIGOJ, A. (1994), "Partitioning Networks on Generalized Concepts of Equivalence," *Journal of Mathematical Sociology*, 19, 1-27.
- [15] DOREIAN, P., BATAGELJ, V., and FERLIGOJ, A. (2001), *Generalized Blockmodeling* (book manuscript) Universities of Ljubljana and Pittsburgh.
- [16] DOREIAN, P. and MRVAR, A. (1996), "A Partitioning Approach to Structural Balance" *Social Networks*, 18, 149-168.
- [17] EVERETT, M.E and BORGATTI, S.P. (1996), "Exact Colorations of Graphs and Digraphs" *Social Networks*, 18, 319-331.
- [18] FAUST, K. (1988), "Comparison of Methods for Positional Analysis: Structural and General Equivalences," *Social Networks*, 10, 313-341.
- [19] FRENCH, C (1963), "Some Structural Aspects of a Retail Sales Group" *Human Organization*, XXII, 146-151.
- [20] HARARY, F., NORMAN, R. Z., and CARTWRIGHT, D. (1965), *Structural Models: An Introduction to the Theory of Directed Graphs*, New York: John Wiley & Sons.
- [21] JENNINGS, H. H.(1948), *Sociometry in Group Relations: A Manual for Teachers* , Westport: Glenwood Press.
- [22] LORRAIN, F., and WHITE, H. C. (1971), "Structural Equivalence of Individuals in Social Networks," *Journal of Mathematical Sociology*, 1, 49-80.
- [23] PATTISON, P. (1988), "Network Models; Some Comments on Papers in this Special Issue," *Social Networks*, 10, 383-411.
- [24] SAILER, L. D. (1978), "Structural Equivalence: Meaning and Definition, Computation and Application," *Social Networks*, 1, 73-90.
- [25] WHITE, D. R., and REITZ, K.P. (1983), "Graph and Semigroup Homomorphisms on Networks of Relations," *Social Networks*, 5, 193-234.