Journal of Classification 8:65-91 (1991)

A Permutation-Based Algorithm for Block Clustering

Diane E. Duffy

Bellcore

Adolfo J. Quiroz

Universidad Simon Bolivar

Abstract: Hartigan (1972) discusses the direct clustering of a matrix of data into homogeneous blocks. He introduces a stepwise divisive method for block clustering within a certain class of block structures which induce clustering trees for both row and column margins. While this class of structures is appealing, the stopping criterion for his method, which is based on asymptotic theory and the assumption that the individual elements of the data matrix are normally distributed, is quite restrictive. In this paper we propose a permutation-based algorithm for block clustering within the same class of block structures. By using permutation arguments to decide where to split and when to stop, our algorithm becomes applicable in a wide variety of cases, including matrices of categorical data and matrices of small-to-moderate size. In addition, our algorithm offers considerable flexibility in how block homogeneity is defined. The algorithm is studied in a series of simulation experiments on matrices of known structure, and illustrated in examples drawn from the fields of taxonomy, political science, and data architecture.

Keywords: Binary splitting; Block clustering; Markov chain simulation method; Permutation distribution.

We would like to thank our many colleagues at Bell Communications Research who have listened to, encouraged, and commented on various aspects of this work, especially Ed Fowlkes, George Furnas, Joop Kemperman, Jon Kettenring, and Debby Swayne. We would also like to thank David Pollard for helpful discussions concerning the proof in Appendix 2, and the editor and three referees for their careful review and for numerous suggestions that significantly improved the presentation of the material. Lastly, our deepest appreciation goes to Debby Swayne who developed all the graphics software required to produce Figures 2-5.

Authors' Addresses: Diane E. Duffy, Bellcore, Morristown, NJ, USA and Adolfo J. Quiroz, Universidad Simon Bolivar, Caracas, Venezuela.

1. Introduction

A block clustering problem is one in which a data matrix is to be partitioned into homogeneous rectangular blocks after reordering the rows and the columns. The objectives of the analysis are to identify blocks or clusters of similar data values, to identify clusters of similar rows and columns, and to explore the relationships between the marginal (i.e., row and column) clusters and the data blocks. Besides block clustering, this technique has also been called block modeling, and direct or two-way clustering in the literature (Arabie, Boorman and Levitt 1978; Hartigan 1972, 1975, 1976).

Data amenable to block clustering methods arise in many fields including taxonomy (Hartigan 1972; Sokal and Sneath 1965), political science (Deutsch and Martin 1971; Hartigan 1976), ecology (Lambert and Williams 1962; Schmid 1984), and business (Breiger, Boorman and Arabie 1975). In a simple taxonomy example, the rows of the matrix concern species, the columns concern characteristics, and the matrix entries denote whether, or to what extent, the species corresponding to that row exhibits the characteristic corresponding to that column.

A wide variety of procedures have been proposed for finding patterns in data matrices. The procedures differ in the patterns they seek, the types of data to which they apply, and the assumptions on which they rely. (See Adelson, Norman and LaPorte 1976; Arabie and Boorman 1982; Bock 1979; Breiman, Friedman, Olshen and Stone 1984; De Soete, DeSarbo, Furnas and Carroll 1984; Gilula 1986; Goodman 1981; Govaert 1977; Greenacre 1988; Heiser and Meulman 1983; Hill 1974; Holland and Leinhardt 1981; Hubert 1974; Hubert and Golledge 1981; Wang and Wong 1987; and Wong 1987.) Our algorithm is a modification of an algorithm introduced in Hartigan (1972) and based on stepwise binary splitting. The next few paragraphs introduce the required terminology and give a brief description of Hartigan's proposal.

At the k-th step of the procedure, the original data matrix has been divided into k blocks (to start, there is one block formed by the matrix as a whole). Either one of the k blocks will be split (by rows or by columns) into two new blocks, or the procedure will stop. In order to decide whether to split and, if so, where, each of the existing blocks is analyzed. Hartigan uses the sum of squares about the mean as a measure of block heterogeneity, and the reduction in sum of squares to assess potential splits. Let $\mathbf{B}$ be an existing block with n rows and m columns; let R (C) be the row (column) set of $\mathbf{B}$ and let b be the mean of the values in $\mathbf{B}$. Suppose $\mathbf{B}$ is split by columns into $\mathbf{B}'$ and $\mathbf{B}''$; $\mathbf{B}'$ and $\mathbf{B}''$ will have the same row set R. Let their column sets, column dimensions and means be C' and C'', m' and m'', and b' and b'', respectively. Note that $C' \cup C'' = C$ and $m' + m'' = m$. The reduction in sum of squares about the mean associated with this split is denoted RSS and

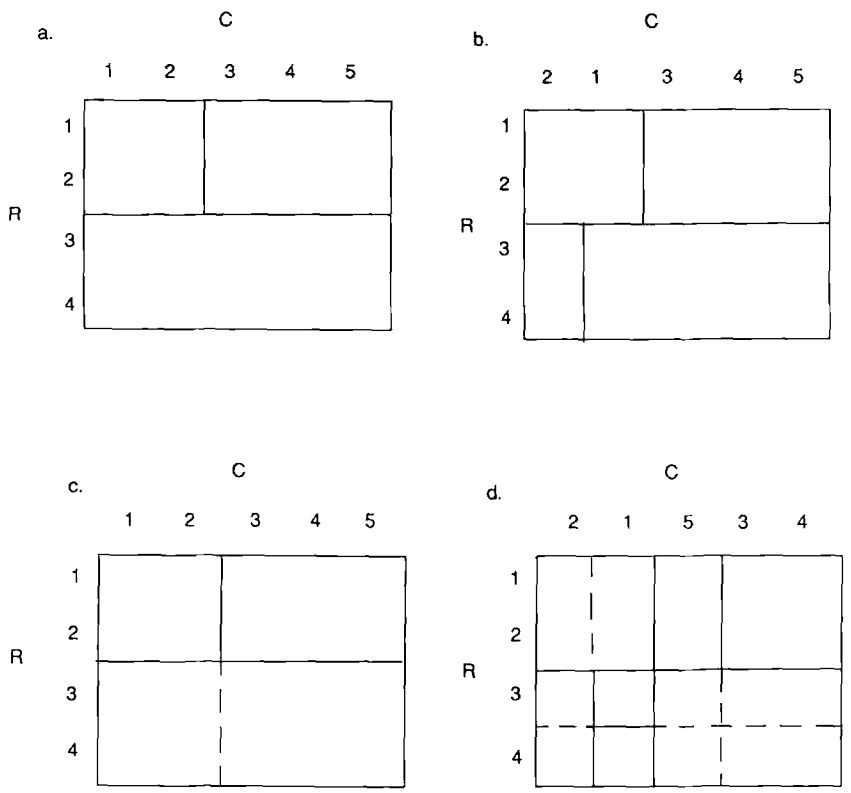

Figure 1. Fixed and Free Splits.

given by

$$RSS = nm'(b' - b)^2 + nm''(b'' - b)^2 \tag{1}$$

Of all possible column splits of **B**, the best split is that one which maximizes *RSS*; similarly for row splits of **B**.

The best row or column split of a given block **B** may result in a structure which can not be contiguously represented following reordering of the rows and columns. For example, consider Figure 1a with $k = 3$.

It is impossible to split the lower block into one subblock consisting of columns 1 and 3 and the other of columns 2, 4, and 5 and represent the structure contiguously. Even when contiguous representation is possible, a split may not permit unambiguous interpretations as partitions on the row and column margins. For example, in Figure 1b the first two rows define the column partition $P_1 = (\{1,2\},\{3,4,5\})$, and the second two rows define the

column partition $P_2 = (\{2\},\{1,3,4,5\})$, and P_1 and P_2 are not hierarchically related. To preserve marginal trees for the rows and columns, Hartigan introduces the notion of "fixed splits." For example, in Figure 1c the dotted line is a fixed split and it is interpreted as implying that if the lower block is split by columns then it must be split at the dotted line. Hence in Figure 1c there is only one allowable column split in the lower block; namely, the fixed split. The best row split of the lower block and the best row and column splits of the upper two blocks are determined in the usual way by maximizing the *RSS* over all binary splits. Figure 1d illustrates four fixed splits at the seventh step as dotted lines. Any split which is not fixed is called a "free split."

Hartigan defines the best split of a block **B** as that split which maximizes a weighted *RSS* where the weighting factor serves to make free and fixed splits comparable. The best overall split is analogously defined as that split which maximizes the weighted *RSS* over all k existing blocks. The procedure stops when

$$SS3/N_3 > \frac{\frac{1}{2}RSS1 + RSS2}{N_1/\pi + N_2}, \tag{2}$$

where N_1 is the total number of rows and columns involved in free splits, N_2 is the total number of fixed splits, N_3 is the total number of entries in the matrix minus the number of blocks, $RSS1$ is the total reduction in sum of squares resulting from free splits, $RSS2$ is the total reduction in sum of squares resulting from fixed splits, and $SS3$ is the total within blocks sum of squares. If inequality (2) does not hold, then the best overall split is performed and the algorithm continues to step $k + 1$. The weighting factor and the stopping criterion are derived by assuming that the entries within each block are independent and identically distributed (iid) according to a normal distribution, and that all of the blocks are large enough to appeal to asymptotic results about the order statistics from Gaussian samples.

The structures produced by this method are appealing because they permit interpretation of the blocks according to row and column partitions. However, the reliance on asymptotic normal theory limits the validity of the approach to certain matrices of numerical data. We conclude this section with a brief description of the data set which motivated our research — one for which the normality assumption is clearly inappropriate. In Section 2 we describe our algorithm giving particular attention to the motivation behind the permutation distribution and to the use of this distribution to define a rule for selecting the best overall split and a stopping criterion. Section 3 illustrates the method on several real examples, and Section 4 reports the results of simulation experiments designed to assess the algorithm's performance on matrices of known structure. The final section indicates some areas of current and future research.

Example 1: The following example from the field of data architecture design is discussed in detail in Duffy, Fowlkes and Kane (1987). At the top or strategic level, data are viewed as a valuable corporate resource and the goals of data architecture design are to use this resource efficiently by organizing data into subject databases. As part of a strategic design effort for a generic operating telephone company, a data matrix consisting of 99 rows corresponding to high-level functions and 23 columns corresponding to high-level objects was created. Examples of functions are 'service order formulation', 'circuit status reporting', and 'vendor invoice processing'; examples of objects are 'customer', 'circuit', and 'service.' The entries in the matrix were 1 or 0 according to whether the function associated with that row did or did not require access to information about the object corresponding to that column during execution. One of the questions of interest was to identify and interpret clusters of homogeneous blocks within this matrix. Figure 2 shows the original function-object interaction matrix with ones represented by solid dots and zeros represented by blanks.

2. A Permutation-Based Block Clustering Algorithm

In this section we describe our block clustering algorithm. The rationale behind the permutation distribution approach is discussed first, followed by the stopping rule and the split selection criterion. The section concludes with some comments on implementation issues.

2.1 The Permutation Distribution as a Reference

For ease of exposition, the following discussion is phrased in terms of free column splits of a block **B**; the analysis of free row splits is completely analogous. Fixed splits are handled similarly, as explained below. Given a heterogeneity reduction measure (e.g., the reduction in sum of squares about the mean) let λ_0 denote the best (largest) reduction attainable by a column split of **B**. In order to assess the value of λ_0, we need to know what values of the reduction measure could be expected under a null model of no structure in block **B**. Hartigan used an iid Gaussian model with large block size as the null model; we would like an alternative null model that does not require either parametric assumptions about the entries or large block size. Our null model assumes that, conditionally on the entries present in the block **B**, all possible permutations of these entries within the block are equally likely. This assumption is sometimes called exchangeability, and reflects the idea that any block structure present is captured in the way the entries are arranged in the different rows and columns. With the exchangeability assumption, we can assess the value of λ_0 by determining how large it is with respect to the set of values corresponding to all the permutations of the entries in **B**.

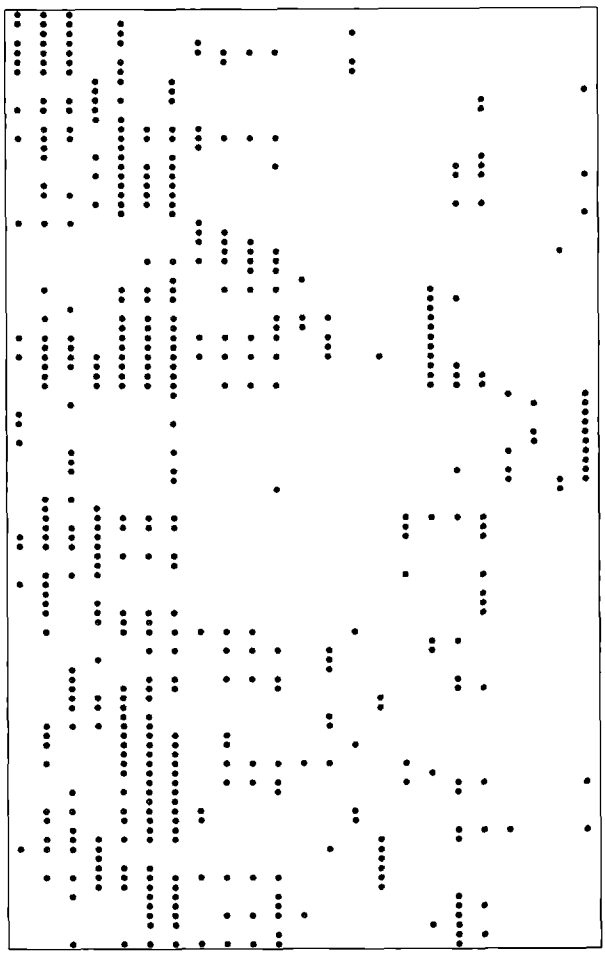

Figure 2. Function-Object Interaction Matrix before Block Clustering.

More specifically, let n and m be the number of rows and columns, respectively, of the block **B**. Let $\mathbf{R} = \mathbf{R}(n,m)$ denote the set of all permutations on $\{(i,j)\colon\ 1 \le i \le n,\ 1 \le j \le m\}$. For each permutation σ in **R**, compute the best reduction in heterogeneity, $\lambda(\sigma)$, that can be obtained with a column split of the block obtained by applying the permutation σ to **B**. Note that $\lambda_0 = \lambda(id)$, where id is the identity permutation. Following the usual approach to permutation-based inference, the more extreme λ_0 is relative to $\{\lambda(\sigma)\colon \sigma \in \mathbf{R}\}$, the more important the associated split is considered. (Note that it is essential for this comparison that $\lambda(\sigma)$ be the best reduction in heterogeneity in the permuted block.) We define the importance of the best column split s of **B**, $\alpha(s)$ as

$$\alpha(s) = Pr(\lambda(\sigma) > \lambda_0) + \tfrac{1}{2} Pr(\lambda(\sigma) = \lambda_0), \quad (3)$$

where σ is uniformly sampled from $\mathbf{R}$. For a fixed column split s of block $\mathbf{B}$, which divides the column set C of B into C' and C'', the calculation of $\alpha(s)$ is similar. Again, σ is sampled uniformly from $\mathbf{R}$, but now $\lambda(\sigma)$ is defined as the reduction in heterogeneity for the given fixed split s on the rearranged data. With this definition of $\lambda(\sigma)$, the importance of the split is again given by (3).

There is one situation in which the permutation method can not be used to compute the importance value for a split. This situation arises when we consider the free split of a block that has only one row or only one column. In this case, the values $\lambda(\sigma)$ are constant over all permutations in $\mathbf{R}$. If the data are numerical and our measure is the reduction in sum of squares, one (non-parametric) way to assess the value of λ_0 is to think of the data in the block as coming from a probability distribution and test that distribution for bimodality by comparing the observed value of λ_0 with the largest value of reduction in sum of squares that can be expected from a unimodal distribution. The rationale for this procedure is the intuition that the reduction in sum of squares is expected to be smaller for unimodal distributions than for bimodal distributions. It can be shown (see Appendix 1), that if $X_1, \ldots, X_n$ are an iid sample from a unimodal density f supported on an interval $[a,b]$ and if the measure of interest is the largest reduction in sum of squares attainable by dividing the sample into two subsamples, then the measure is asymptotically maximized (as $n \to \infty$) when f is a mixture of a point mass at a and the uniform distribution on $[a,b]$. Therefore, one can use the mixture distribution as a reference to evaluate the magnitude of λ_0. In our implementation, for the sake of simplicity, we use the uniform distribution on $[a,b]$ as the reference, estimating the values of a and b from the data in the block, by their maximum likelihood estimators. In the case of categorical data, the problem of $n \times 1$ and $1 \times m$ blocks is handled similarly, by using as a reference the uniform distribution on the set of modes present in the block.

The assumption of exchangeability and the ensuing permutation-based inference are commonly used tools in statistics for deriving procedures which are not dependent upon parametric assumptions. Besides the natural appeal of this approach in the context of block clustering, it also has the advantage of allowing the use of measures of reduction of heterogeneity other than the reduction in sum of squares. This increased flexibility further widens the domain of applicability of the permutation-based algorithm.

For example, with categorical data taking values in the set $\{1, \ldots, r\}$, let $f(\mathbf{B}) = (f_1(\mathbf{B}), \ldots, f_r(\mathbf{B}))$ be the frequency vector of values $1, \ldots, r$ in block $\mathbf{B}$; that is, $f_i(\mathbf{B})$ = (number of entries in $\mathbf{B}$ with value i) $/ nm$, and $\sum_{i=1}^{r} f_i(\mathbf{B}) = 1$. If split s divides $\mathbf{B}$ into $\mathbf{B}'$ and $\mathbf{B}''$, one heterogeneity reduction

statistic based on the discrepancy between $f(\mathbf{B}')$ and $f(\mathbf{B}'')$ is

$$\lambda(s) = w(\mathbf{B}', \mathbf{B}'') \sum_{i=1}^{r} |f_i(\mathbf{B}') - f_i(\mathbf{B}'')|, \tag{4}$$

where $w(\mathbf{B}', \mathbf{B}'')$ is a nonnegative weight. When $w(\mathbf{B}', \mathbf{B}'') = 1$, (4) agrees with both the total variation and Prohorov (Billingsley 1968, p. 238) distances between $f(\mathbf{B}')$ and $f(\mathbf{B}'')$ when they are viewed as discrete probability mass functions on $1, \ldots, r$ (see Appendix 2).

2.2 The Stopping Criterion

At the k-th step of the algorithm, let Q_k contain the best free splits by column and row of all blocks plus all fixed splits. For each split s in Q_k, suppose that the importance level $\alpha(s)$ has been computed according to the recipe given in the previous section.

We stop splitting when there is no split s in Q_k with

$$\alpha(s) \leq \alpha_k \tag{5}$$

where the sequence $\{\alpha_1, \alpha_2, \ldots,\}$ of cutoff points is given *a priori*. The sequence α_k should decrease as $k \to \infty$ because with larger k there are more possible blocks to split. The rate at which α_k decreases controls the degree of fine structure in the resulting block pattern. The choice $\alpha_k = .5^k$ reflects the concept that for k independent blocks the probability of any of the k blocks yielding a split that is better than expected by chance is $1 - 2^{-k}$.

We have experimented with other geometric sequences of the form $\alpha_k = \alpha_0^k$, for α_0 in [.5, .75], and suggest that in applications a few choices of α_0 be tried. There are two major reasons for favoring geometrically decreasing α_k (over other sequences decreasing more slowly). First, in most applications it is the large trends and patterns in the block structure that are particularly illuminating, and second, the larger the α_k, the greater the possibility that fine patterns in the block structure result from chance.

2.3 Split Selection

If the stopping criterion is not met, then at least one of the splits in Q_k satisfies (5). Of the splits satisfying (5) we execute the one with the largest value of $\lambda(s)$, with ties broken arbitrarily. That is: of the splits with enough importance, we execute the one that maximizes the reduction in heterogeneity.

2.4 Implementation Issues

Two implementation issues will be briefly discussed here: first, finding the best free split by row or column of a given block **B**, and second, calculation of the importance level $\alpha(s)$.

2.4.1 Finding the Best Free Split

Suppose that we are looking for the best free column split s of an $n \times m$ block **B** into blocks $\mathbf{B}'$ and $\mathbf{B}''$. In principle, one should consider all $2^{m-1} - 1$ possible splits of the column set C, of **B**. However, in certain cases, one can analyze a considerably smaller set of splits. If the data are numerical and the measure of heterogeneity being used is the sum of squares about the mean, then Hartigan (1972) showed it is sufficient to order the columns of **B** in increasing order of column sum, and consider only the $m - 1$ possible splits which preserve this order. (Note that ties between column sums can be broken arbitrarily since a tie leads to two splits with identical values of the reduction statistic and hence identical importance levels.) For matrices of categorical data, when the measure of reduction in heterogeneity is given by (4), and when the number of modes is two, a similar reduction in computation is possible. In this case it suffices to order the columns in increasing order of occurrence of one of the modes. The best column split will again be one of the $m - 1$ possible splits of the ordered columns. To prove this, it is enough to see that when there are two modes the formula in (4) reduces to

$$\lambda(s) = 2w(\mathbf{B}', \mathbf{B}'')\,|f_1(\mathbf{B}') - f_1(\mathbf{B}'')|\,, \tag{6}$$

and then notice that for each value of m, this expression is maximized when the columns with the fewest occurrences of mode 1 are on the same side of the split.

For numerical data, when measures of heterogeneity different from the sum of squares are used (the sum of absolute deviations around the median, for example), the method of ordering the columns by sums and then considering only $m - 1$ splits does not necessarily provide the largest reduction in heterogeneity. Similarly, when the frequency-based measure given in (4) is used and the number of modes r is greater than two, there does not seem to be a way to order the columns so as to guarantee finding the best column split after evaluating only $m - 1$ splits. To reduce the computation, a lexicographic ordering technique can be used which does not necessarily find the best split, but hopefully finds a split that is close to best. This method involves ordering the columns lexicographically with respect to the alphabet consisting of the r modes in decreasing order of their frequency within block

B. For example, if $r = 3$, $n = 5$, $m = 3$, and the three columns of the block are: column 1 = [2 1 1 2 2], column 2 = [3 1 2 3 2], and column 3 = [1 2 1 2 1], then the frequency alphabet is 2,1,3 because 2 is the most frequently observed mode (7 occurrences) and 3 is the least frequently observed mode (2 occurrences). Ordering the columns lexicographically with respect to this alphabet, we have column 1 first, column 3 next, and column 2 last. (In this small example the best column split, which separates columns 1 and 3 from column 2 and has value .8 for measure (4), does respect the lexicographic ordering; this will not always happen.)

2.4.2 Calculation of $\alpha(s)$

Once the reduction in heterogeneity λ_0 due to a free or fixed split s of the block **B** has been found, we need to compute $\alpha(s)$ in order to assess the potential split. One way to approach the problem is to look at (5) as a decision problem; sample permutations from **R** and compute the respective values of $\lambda(\sigma)$ until there is enough evidence either to reject or accept (5). Another method of attack is to obtain an approximate value for $\alpha(s)$ through simulation. A straightforward Monte Carlo implementation involves sampling uniformly from **R**, and computing $\lambda(\sigma)$ for each sampled permutation σ. In contrast, a Markov chain simulation method considers a Markov chain with state space **R** and transitions defined by transposing two randomly-chosen entries of **B**; that is, if we start at state σ_1 (uniformly randomly chosen from **R**), then state σ_2 either differs from σ_1 by having exactly two entries transposed or $\sigma_2 = \sigma_1$. (The latter occurs if the entries chosen to be transposed were identical.) For each state σ_i in the Markov chain, we compute the corresponding value $\lambda(\sigma_i)$ and keep a count of how many of these are smaller than λ_0 in order to decide whether (5) holds. We run the Markov chain long enough so that the probability of being in any state approaches $1/|\mathbf{R}|$ where $|\mathbf{R}|$ denotes the cardinality of set **R**. Bounds for the number of steps necessary for the distribution of σ_i, the i-th permutation observed in the Markov chain, to be within a given distance of the uniform distribution on **R** have been given by Diaconis and Shashahani (1981). Using their results as guidelines, we let the Markov chain on **R** run for $50 + 10\,N \log(N)$ steps, where $N = mn$ is the size of the block **B**.

The use of Markov chains to simulate uniform distributions over combinatorial sets has been around for a while. Aldous (1987) mentions some of the early applications. The key notion is that a Markov chain is set up with transitions corresponding to minimal changes (i.e., transpositions) that can be made to any element of the state space. Aldous also addresses the question of how long the chain must be run to guarantee that the observed average of a real-valued function defined on the state space is close to the actual mean of

the function under the uniform distribution. Aldous considers a much more general class of combinatorial sets and takes an algorithmic approach which leads to conditions which guarantee polynomial time accuracy.

In our context, the use of the Markov method has the advantage that the calculations necessary to compute $\lambda(\sigma_i)$ can be used to speed the calculation of $\lambda(\sigma_{i+1})$; for instance, if the columns of the block are ordered (by column sum) with respect to σ_i, then they are 'almost ordered' with respect to σ_{i+1}. The fact that we are simulating a realization of the Markov chain via random transpositions to get the value of $\alpha(s)$ makes the outcome of our algorithm random. Our empirical study of the behavior of the algorithm (Section 4) shows that the variability of the output of the algorithm is very small; our examples (Section 3) show how this slight variability can be used to help assess the structures in a data matrix.

3. Examples

In this section three examples are analyzed with the block clustering algorithm. For each example, several choices of α_0, the fraction governing the geometric cut-off sequence, were tried and, for each choice of α_0, the algorithm was run five times with different seeds governing the Markov chain simulation method.

Example 1 (cont.) Figure 3 shows the results of applying the permutation-based algorithm with heterogeneity reduction given by (1) and $\alpha_0 = .75$ to the function-object interaction matrix from the data architecture design problem. Again, ones are represented in the figure by solid dots and zeros are represented by blanks. The marginal trees are drawn in the right and bottom margins of the figure. In this clustering there are 15 blocks with four row clusters and eight column clusters. For the 14 splits performed, the values of the reduction statistic (1) and the importance $\alpha(s)$ were (19.1, 0), (43.0, 0), (10.9, 0), (10.9, 0), (4.4, 0), (7.6, 0), (.7, .05), (18.3, 0), (2.0, 0), (.9, 0), (.5, 0), (2.8, .03), (1.0, 0), and (1.3, 0) respectively, for the run plotted. In general, it is our experience that over half the $\alpha(s)$ values on a given run are zero, implying that no rearrangement yielded a split with as large a value of the reduction statistic.

Figure 3 depicts a very large and complex set of data. Although considerations of length and comprehensibility preclude a detailed discussion of its analysis, we would like briefly to comment on several points. First, we want to point out that the data analyzed in this example motivated our research. Procedures based on assuming normality are clearly inappropriate because of the binary nature of the data; the permutation approach, on the other hand, proved both appropriate and practical. Our algorithm divides the

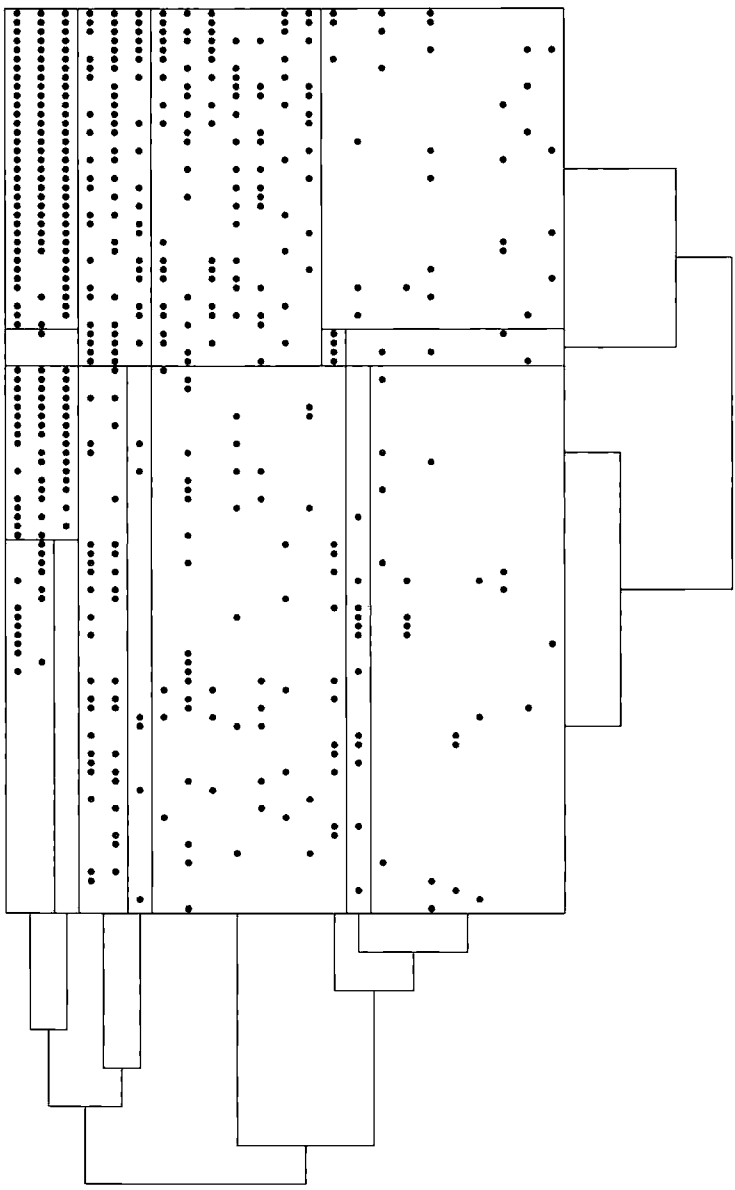

Figure 3. Function-Object Interaction Matrix After Block Clustering.

matrix into blocks which differ according to the density of ones, and which highlight some of the key relationships in the data. For example, the first three columns correspond to the product, assignment inventory, and sub-assembly inventory objects, all of which are concerned with the logical parts

that comprise the telephone network. The upper left block identifies a large group of functions which need access to most of these objects. Just below the upper left block is a small block with only one solid dot. The rows in this block correspond to four functions which all deal with the circuit which provides service to a customer. These four functions, with one exception, do not need to know about the lower level of objects which comprise the circuit. One way to assess our approach is to compare Figure 3 with Figure 2. The display in Figure 2 is the best solution that the subject-matter experts were able to come up with when they tried to uncover block structure by reordering the rows and columns of the matrix by hand.

Example 2. Hartigan (1976) presents data on the dentition of 32 West Coast mammals. For each mammal, the numbers of top and bottom canines (TCAN, BCAN), top and bottom incisors (TINC, BINC), top and bottom premolars (TPRM, BPRM), and top and bottom molars (TMOL, BMOL) are given, producing a 32×8 matrix. The 32 mammals come from seven phylogenic orders: Marsupalia (abbreviated M; 1 species represented), Insectivora (I; 4), Chiroptera (CH; 9), Lagomorpha (L; 1), Rodentia (R; 4), Carnivora (CR; 10), and Artiodactyla (A; 3). One reason to include this example is that there is an objective standard to compare to the row partition, namely, the phylogenic classifications. Of course, we would not hope to reproduce phylogenic classifications based on dentition data alone, but we would expect some correspondence between our row partition and the phylogenic orders.

Figure 4 shows the results of applying the algorithm with heterogeneity reduction given by (1) and $\alpha_0 = .70$. The two dotted lines in Figure 4 denote the following: in repeated runs with different seeds, either the dotted split on the left associated with rows RABBIT and GREY SHREW, and columns TCAN and BCAN was performed, or the two dotted splits on the right associated with rows LYNX, MOUNTAIN LION, and SEA OTTER were performed. It was never the case that all three dotted splits were performed because doing so would violate the row partition. Thus, doing several runs with different seeds not only aids in assessing the stability of the clustering, but can also point out additional features. For the 13 solid splits shown, the values of the reduction statistic (1) and the importance (3) for the run plotted were (129.2, 0), (26.2, .19), (15.6, .003), (16.9, .003), (4.12, 0), (18.6, 0), (7.5, .014), (1.5, 0), (2.8, 0), (2.0, 0), (7.4, 0), (2.0, 0), and (1.6, 0), respectively. For the three dotted splits, the values were (1.8, 0), (14.1, 0), and (5.0, .03).

In Figure 4, the 32 mammals are divided into eight groups:

Group 1: MOUSE, JUMPING MOUSE, GOPHER, SQUIRREL.
Group 2: PALLID BAT.
Group 3: RABBIT, GREY SHREW.

	TCAN	BCAN	TINC	BINC	TPRM	BPRM	TMOL	BMOL
MOUSE (R)	0	0	1	1	0	0	3	3
JUMPING MOUSE (R)	0	0	1	1	1	0	3	3
GOPHER (R)	0	0	1	1	1	1	3	3
SQUIRREL (R)	0	0	1	1	2	1	3	3
PALLID BAT (CH)	1	1	1	2	1	2	3	3
RABBIT (L)	0	0	2	1	3	2	3	3
GREY SHREW (I)	1	0	3	2	1	1	3	3
LONG-TONGUED BAT (CH)	1	1	2	0	2	3	3	3
FREE-TAILED BAT (CH)	1	1	1	2	2	2	3	3
BROWN BAT (CH)	1	1	2	3	1	2	3	3
RED BAT (CH)	1	1	1	3	2	2	3	3
SHREW (I)	1	1	3	1	3	1	3	3
LYNX (CR)	1	1	3	3	2	2	1	1
MOUNTAIN LION (CR)	1	1	3	3	3	2	1	1
SEA OTTER (CR)	1	1	3	2	3	3	1	2
SEA LION (CR)	1	1	3	2	4	4	1	1
FUR SEAL (CR)	1	1	3	2	4	4	2	1
RIVER OTTER (CR)	1	1	3	3	4	3	1	2
MINK (CR)	1	1	3	3	3	3	1	2
MARTEN (CR)	1	1	3	3	4	4	1	2
RACCOON (CR)	1	1	3	3	4	4	2	2
WOLF (CR)	1	1	3	3	4	4	2	3
PIPISTRELLE (CH)	1	1	2	3	2	2	3	3
ELK (A)	1	1	0	3	3	3	3	3
LEAF-NOSED BAT (CH)	1	1	2	2	2	3	3	3
SHREW MOLE (I)	1	1	3	3	2	2	3	3
MYOTIS (CH)	1	1	2	3	3	3	3	3
SILVER-HAIRED BAT (CH)	1	1	2	3	2	3	3	3
DEER (A)	0	1	0	3	3	3	3	3
GOAT (A)	0	1	0	3	3	4	3	3
MOLE (I)	1	1	3	3	4	4	3	3
OPOSSUM (M)	1	1	5	4	3	3	4	4

Figure 4. Number of Teeth by Mammal Species and Tooth type.

Group 4: LONG-TONGUED BAT, FREE-TAILED BAT, BROWN BAT, RED BAT, SHREW.
Group 5: LYNX, MOUNTAIN LION, SEA OTTER.
Group 6: SEA LION, FUR SEAL, RIVER OTTER, MINK, MARTEN, RACCOON.
Group 7: WOLF, PIPISTRELLE, ELK, LEAF-NOSED BAT, SHREW MOLE, MYOTIS, SILVER-HAIRED BAT, DEER, GOAT, MOLE.
Group 8: OPOSSUM

For most species, there are distinctions between the number of canines, molars, and other teeth, and this situation is reflected in the two series of splits which completely separate the columns into canines, molars, and others. Of the six blocks on the right involving molars, two serve to help identify the carnivores by the few molars they have, one serves to help identify the marsupial by its large numbers of molars, and the remaining three are, with the exception of one entry, homogeneous blocks of 3's. Of the five blocks on the left involving canines, the first block helps identify the rodents by the fact that they have no canines; most of the rest of the species have one each of top and bottom canines. The rodents are further identified by their small counts of incisors and premolars. The carnivores with one exception (WOLF) fall into Groups 5 and 6 and are distinguished by having a total of four or fewer molars. The carnivores in Group 5 have slightly fewer molars and slightly more incisors and premolars than those in Group 6. The opossum, the only marsupial, is identified by its large number of top incisors and molars. The one-column block formed by the column corresponding to TINC and the rows corresponding to Groups 6 and 7 has been split off primarily because of its broader range of values (0 to 3) compared to the values in the neighboring block formed by the same rows but by the columns corresponding to BINC, TPRM, and BPRM.

Example 3. Hartigan (1972) presents data on votes cast at the United Nations during 1969-1970; 19 countries and 14 resolutions are considered. He gives brief descriptions (1972, p. 125) of the resolutions; we have used his numbering scheme to label the resolutions with added words to indicate the subject. The votes are coded: 1 = Yes, 2 = Abstain, 3 = No, and 4 = Absent. Figure 5 presents the clustering obtained with the frequency-based heterogeneity reduction (4) and $\alpha_0 = .70$. It should be noted that our implementation of reduction measure (4) for this case of $r = 4$ modes relied on the lexicographic ordering mentioned in Section 2.4 to reduce the computation. This ordering is independent of the way in which the four modes are coded. The exact clustering in Figure 5 was obtained each of the five times that the algo-

	1 - KOREA	10 - KOREA	6 - CHINA	13 - PAPAU	5 - CHINA	12 - S. AFRICA	11 - S. AFRICA	7 - CHINA	14 - PAPAU	3 - HUNG	4 - HUNG	2 - HUNG	9 - KOREA	8 - CHINA
FRANCE	1	1	1	2	3	3	1	2	2	3	3	3	3	3
SWEDEN	1	1	1	3	3	3	1	2	1	3	3	3	3	3
ALBANIA	1	1	3	3	1	3	1	1	1	3	3	3	3	3
UNITED ARAB REP	1	2	1	1	3	1	3	1	3	3	3	3	2	3
KENYA	1	4	1	1	3	1	3	1	3	3	3	3	2	3
UNITED KINGDOM	1	1	1	3	1	3	1	3	1	3	3	3	3	2
NORWAY	1	1	1	3	3	3	1	3	1	3	3	3	3	2
USA	1	1	3	3	1	3	1	3	1	3	3	3	3	1
YUGOSLAVIA	1	2	1	1	3	1	3	1	2	3	3	3	1	3
NEW ZEALAND	1	1	3	3	1	3	1	1	1	3	3	3	3	1
DAHOMEY	1	1	3	2	1	1	3	1	2	3	3	3	4	3
SENEGAL	1	1	2	1	1	1	3	2	2	3	3	3	2	2
TANZANIA	1	4	1	1	3	1	3	1	3	2	2	2	2	3
SYRIA	1	2	1	1	3	1	3	1	3	2	2	2	1	3
MEXICO	1	1	3	2	1	1	1	3	1	2	2	2	3	1
VENEZUELA	1	1	3	1	1	1	2	3	1	2	2	2	3	1
BRAZIL	1	1	3	1	1	3	1	3	1	2	2	2	3	1
USSR	1	3	1	1	3	2	2	2	3	1	1	1	1	3
BULGARIA	1	3	1	1	3	2	2	1	3	1	1	1	1	3

Figure 5. UN Votes by Country and Resolution.

rithm was run with different seeds. For the run plotted, the values of the reduction statistic (4) and the importance (3) for each of the six splits were (3.8, 0), (4.7, 0), (3.5, .003), (3.9, 0), (2.9, .02), and (2.7, 0), respectively.

The main feature of Figure 5 is the presence of four completely homogeneous blocks. The first homogeneous block includes all countries and corresponds to a resolution calling for eased tensions in Korea. This was apparently such a vague, innocuous resolution that everyone could agree on it! The three resolutions titled 2-HUNG, 3-HUNG, and 4-HUNG serve to divide the 19 nations into three homogeneous groups: a large group consisting of western European countries, the United States, and several other countries which voted no on all three resolutions; a medium-sized group consisting of three Latin American countries plus Syria and Tanzania which abstained; and a small group consisting of the Soviet Union and Bulgaria

which voted yes. These three resolutions concern a series of changes, called the Hungarian amendments, to a statement which would expel South Africa from the UNCTAD (United Nations Conference on Trade and Development).

4. Simulation Experiments

In this section we describe the results of a series of simulation experiments designed to study the behavior of the algorithm on matrices of known structure. We distinguish between two types of experiments: null experiments in which the data matrices are generated with iid entries, and non-null experiments in which the data matrices are generated with a specific block structure in mind. The null experiments are designed to explore the degree to which the algorithm "finds" block structure only because the algorithm sets out looking for such structure. We believe that this is an important question which should be asked of any clustering procedure. The null matrices generated will, generally, have no block structure (although by chance some null matrices will have fairly definite block patterns), and the algorithm should terminate quickly after finding very few blocks (ideally, none). Null experiments are denoted B1 for one block.

In contrast, the non-null experiments explore the ability of the algorithm to uncover an intentional pattern of blocks. (Again, because of the random generation mechanism used, some matrices will exhibit the intended pattern more clearly than others.) Here we discuss results for two very simple block patterns: a 2-block pattern denoted B2 and a 4-block pattern denoted B4 (see Figures 6a and b). For each non-null experiment we consider two levels of separation between the blocks representing an easier and a harder case.

More specifically, Table 1 lists the various cases considered and a 5-letter code for each case. The entries in each cell of the table are (i) the parameter settings for blocks 1,2,... and (ii) the code consisting of one letter denoting the distribution followed by 'B1', 'B2' or 'B4' for the type of experiment and ending with 'S1' or 'S2' for the degree of separation. (See Figures 6a and b for the numbering and position of blocks in the non-null cases.) In the non-null experiments S1 represents blocks that are more separated and easier to distinguish; in the null experiments S1 represents the lower variance case. Note that for continuous responses, two distributions are considered, while for binary responses we need only consider the Bernoulli distribution. For each case (i.e., each cell of Table 1) we generated four matrices of the given type with dimension 24×12. The algorithm was then run five times on each matrix with different seeds for the Markov chain simulation method; for uniform and normal data we used heterogeneity reduction (1) while for Bernoulli data we used (4). In all cases $\alpha_0 = .5$.

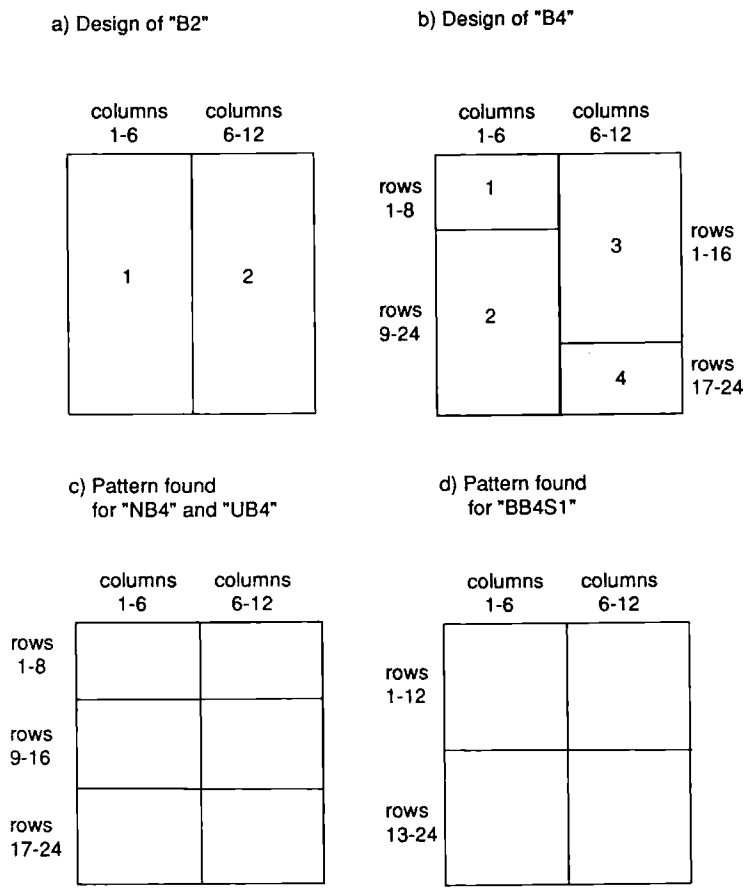

Figure 6. Designs and Patterns for Simulation Experiments

4.1 Summary of Results

Table 2 provides some summary measures of the algorithm's performance. Note that each row of this table is based on 20 applications of the associated algorithm — four matrices of the associated type times five runs per matrix. Columns 2 through 4 of Table 2 give the minimum, maximum, and average number of splits. Column 5 gives the number of times (out of four matrices) that all five runs of the algorithm produced the exact same result, and Column 6 gives the difference between the average number of splits performed by the algorithm and the number of splits designed (i.e., the difference between Column 4 and third character in the problem code minus one).

Distribution	Null		Non-Null			
	B1		B2		B4	
Uniform (a,b) =	(0,2)	:UB1S1	(0,2) (2,4)	:UB2S1	(0,1) (1,2) (2,3) (2,4)	:UB4S1
	(0,5)	:UB1S2	(0,1.5) (1,2.5)	:UB2S2	(0,1.5) (1,2.5) (2,3.5) (3,4.5)	:UB4S2
Normal (μ,σ) =	(0,1)	:NB1S1	(0,1) (1,1)	:NB2S1	(0,1) (2,1) (4,1) (6,1)	:NB4S1
	(0,2)	:NB1S2	(0,1) (1,1)	:NB2S2	(0,1) (1,1) (2,1) (3,1)	:NB4S2
Bernoulli p =	.25	BB1S1	.1 .9	:BB2S1	.05 .35 .65 .95	:BB4S1
	.5	:BB1S2	.25 .75	:BB2S2	.15 .40 .60 .85	:BB4S2

Table 1

Design of Simulation Experiments

Focusing first on the null experiments, we see that the average number of splits ranges from .5 to 1.9. The overall average (for all null experiments = 'B1') is 1.37 with no systematic effects resulting from distribution (uniform = 'U' vs. normal = 'N' vs. Bernoulli = 'B') or variability level ('S1' vs. 'S2').

Considering the 2-block case ('B2') the average number of splits ranges from 1.60 to 2.60. The minimum number of splits is one in all cases and the intended split was *always* the first split performed. The overall average number of splits is 1.92 with the averages over the distribution types being 1.78 for 'U', 1.68 for 'N', and 2.30 for 'B'.

Lastly, for the 4-block ('B4') examples, the range of average number of splits is 1.55 to 6.25 with a striking difference between the categorical cases (Bernoulli) with an average of 2.45 and the continuous cases (uniform and

Problem Code	Min. No. of Splits	Max. No. of Splits	Avg. No. of Splits	Number Fully Consistent	Difference of Avg. No. of Splits Performed to No. of Splits Designed
UB1S1	0	3	1.35	2	1.35
UB1S2	1	3	1.65	3	1.65
NB1S1	0	4	1.75	3	1.75
NB1S2	0	2	1.00	4	1.00
BB1S1	0	1	0.55	2	.55
BB1S2	1	3	1.90	2	1.90
UB2S1	1	3	1.60	3	.60
UB2S2	1	4	1.95	3	.95
NB2S1	1	4	1.75	1	.75
NB2S2	1	3	1.60	2	.60
BB2S1	1	4	2.60	2	1.60
BB2S2	1	4	2.00	3	1.00
UB4S1	5	6	5.10	2	2.10
UB4S2	6	7	6.25	4	3.25
NB4S1	5	6	5.30	3	2.30
NB4S2	2	7	5.50	2	2.50
BB4S1	3	4	3.35	3	.35
BB4S2	1	2	1.55	3	-1.45

Table 2
Results of Simulation Experiments

normal) with an average of 5.54. In the uniform and normal cases, the structure found is basically that shown in Figure 6c. In the Bernoulli case with greater separation ('S1'), the algorithm performs an average of 3.35 splits (usually as shown in Figure 6d), but when the separation is small ('S2') the algorithm performs on average one less split and misses part of the intended pattern.

Column 5 of Table 2 addresses the consistency of the results with respect to the random numbers governing the Markov chain simulation method. For 65% of the matrices generated, five runs with different seeds produced *identical* results (full consistency). For almost all the other matrices, the results were hierarchically consistent in that runs would differ only by the addition of a split or two.

It is worth noting that the randomness inherent in the Markov chain simulation method can be used informally to assess the 'strength' of the block structure. For example, when two hierarchically related solutions occur, then the common splits indicate structure that is strongly present. The occurrence of solutions that differ greatly should lead an analyst to reconsider the appropriateness of block clustering methodology in the given application. In conclusion, the algorithm exhibited good ability to detect the structure present in the matrices and to stop soon afterwards. The algorithm also shows quite credible consistency with respect to the randomness inherent in the Markov chain simulation method.

5. Extensions

The permutation-based algorithm can be extended in several directions. First, the algorithm can be applied with virtually any measure for heterogeneity reduction. Besides the two measures illustrated here, we have done some preliminary work with entropy-based measures for categorical data based on a suggestion due to Joop Kemperman (Duffy and Kemperman 1990). For the entropy measure the best column (or row) split of a block **B** takes the form of a separating hyperplane in the space of frequency vectors $f(\mathbf{B})$, and tools from computational geometry may be helpful in writing an efficient implementation (Edelsbrunner 1987). With continuous responses, obvious alternatives to the reduction in sum of squares about the mean are an L_1-based measure involving summed absolute deviations about the median, or a measure based on a robust estimate of variance. More generally, multidimensional matrix entries can be accommodated if reasonable heterogeneity reduction measures can be derived and implemented.

As presented, our algorithm finds block structures within Hartigan's class of structures which preserve marginal trees for both the rows and columns. We have investigated an alternative maximal class which permits

all binary splits at any time. A major limitation of this maximal class is the difficulty in displaying the results. An intermediate class which restricts to structures which can be displayed as contiguous blocks appears to be a promising area for research.

In order to assess the importance of a split s, we use simulation to calculate an approximation to $\alpha(s)$. Recent advances in statistical computing, specifically, network-based algorithms for exact permutational inference (see Mehta, Patel and Gray 1985 and the references therein), have dramatically reduced the computational demands for some permutation calculations. Extensions of these ideas may make exact calculation of $\alpha(s)$ practical.

The algorithm we propose, like many stepwise techniques, behaves in a "greedy" fashion; at each step the algorithm chooses the optimal course of action without any consideration of how this choice may affect future steps. In particular, the algorithm may stop when perhaps, if it instead performed a less important split at the current step, interesting structure could be found in subsequent splits. One interesting approach to this difficulty would be to permit the algorithm to pursue a less important split with small probability based on an external random mechanism. A more ambitious approach would be to derive an algorithm which looks one step (or, more generally, m steps) ahead before making decisions about splitting or stopping.

Appendix 1
Asymptotic Maximum of RSS for Unimodal Densities

In this Appendix, we show that the reduction in sum of squares from the best split is asymptotically maximized over the space of unimodal densities with support $[a,b]$ when the density is a mixture of a point mass at a and the uniform density on $[a,b]$. The argument proceeds in a step-wise fashion: start with an arbitrary density f and successively define densities based on f which are simpler in structure and for which the reduction in sum of squares is at least as large.

Let F be the family of unimodal densities on $[a,b]$ and, without loss of generality, let $a = 0$, $b = 1$. Consider $f \in F$ and $0 < x < 1$. Asymptotically, for the sum of squares about the mean, calculating $\lambda(s)$ for a split s which divides $[0,1]$ into $[0,x)$ and $[x,1]$ is equivalent to computing

$$R(f,x) = \int_0^1 (t - m_0)^2 f(t)dt - \int_0^x (t - m_1)^2 f(t)dt - \int_x^1 (t - m_2)^2 f(t)dt$$

where

$$m_0 = \int_0^1 tf(t)dt\,,\ m_1 = \frac{1}{p_1}\int_0^x tf(t)dt\,,\ m_2 = \frac{1}{p_2}\int_x^1 tf(t)dt\,,$$

$$p_1 = \int_0^x f(t)dt \quad \text{and} \quad p_2 = \int_x^1 f(t)dt\,.$$

Straightforward calculation shows that

$$R(f,x) = p_1 p_2 (m_1 - m_2)^2\,.$$

Let $x^* = x^*(f)$ be the best split point for f ; i.e., $R^*(f) \equiv R(f,x^*) \geq R(f,x)$ for all $x \in (0,1)$. Our interest focuses on

$$\arg\sup\{R^*(f) : f \in F\}\,;$$

i.e., the density $f \in F$ which maximizes $R^*(f)$. Since the set F^o of bounded continuous unimodal densities is dense in F, it is sufficient to find

$$\arg\sup\{R^*(f) : f \in F^o\}\,. \tag{A.1}$$

Let $f \in F^o$ and without loss of generality assume $x^* = x^*(f)$ is to the right of the mode of f. Define f_1 by:

$$f_1(t) = \begin{cases} f(t)\,, & 0 \leq t \leq x^* \\ a \equiv \dfrac{p_2}{1-x^*}\,, & x^* \leq t \leq 1 \end{cases}$$

Then, $p_1(f_1,x^*) = p_1(f,x^*)$, $p_2(f_1,x^*) = p_2(f,x^*)$, $m_1(f_1,x^*) = m_1(f,x^*)$ and $m_2(f_1,x^*) \geq m_2(f,x^*)$ implying that $R(f_1,x^*) \geq R(f,x^*)$. If the mode of f is at 0, then there exists y, $0 < y < x^*$, such that $f(0)y + a(x^* - y) = p_1$. Define

$$f_2(t) = \begin{cases} f(0)\,, & 0 \leq t < y \\ a & y \leq t \leq 1\,, \end{cases}$$

and note that f_1 and f_2 have identical values of p_1, p_2, and m_2, but $m_1(f_2,x^*) \leq m_1(f_1,x^*)$. Hence $R(f_2,x^*) \geq R(f_1,x^*)$.

If, on the other hand, the mode of f, denoted y, is greater than 0, then define

$$f_2(t) = \begin{cases} b \equiv \dfrac{1}{y}\displaystyle\int_0^y f(u)du\,, & 0 \le t < y \\ f(t)\,, & y \le t < x^* \\ a\,, & x^* \le t \le 1 \end{cases}.$$

Since f is monotonic in $[y,x^*)$, there exists z, $y < z < x^*$ such that

$$f(y)(z-y) + f(x^*)(x^* - z) = \int_y^{x^*} f(t)dt\,.$$

Since $f(y) > f(x^*) \ge a$, the function f_3 defined by:

$$f_3(t) = \begin{cases} b\,, & 0 \le t < y \\ f(y)\,, & y \le t < z \\ f(x^*)\,, & z \le t < x^* \\ a\,, & x^* \le t \le 1 \end{cases},$$

is in F. Further, $R(f_3,x^*) \ge R(f_2,x^*) \ge R(f_1,x^*)$ holds since p_1, p_2, and m_2 are identical for f_1, f_2, and f_3 while $m_1(f_3,x^*) \le m_1(f_2,x^*) \le m_1(f_1,x^*)$. Next let

$$f_4(t) = \begin{cases} c \equiv b + \dfrac{1}{y}\{(f(y)-a)(z-y) + (f(x^*)-a)(x^*-z)\}\,, & 0 \le t < y \\ a\,, & y \le t \le 1 \end{cases}$$

Again, p_1, p_2, and m_2 are constant while m_1 decreases so $R(f_4,x^*) \ge R(f_3,x^*)$.

The previous two paragraphs imply that it is sufficient to find the maximum value of $R(f_5,x^*)$ for densities of the form

$$f_5(t) = \begin{cases} a\,, & 0 \le t < y \\ b\,, & y \le t \le 1 \end{cases}$$

with $0 < y < x^*$. If $a > b$, then we increase the value of R by moving some of the mass of $f(t)$ for $t \le y$ to the left. That is, let $0 < z < y$ and define

$$f_6(t) = \begin{cases} c \equiv b + \dfrac{y}{z}(a-b)\,, & 0 \le t < z \\ b\,, & z \le t \le 1\,, \end{cases}$$

then $R(f_6, x^*) \geq R(f_5, x^*)$. This shifting of mass to the left can be continued implying that $R(f_7, x^*) \geq R(f, x^*) = R^*(f)$ where $f_7 = \alpha\delta_0 + (1-\alpha)U$ with $0 \leq \alpha \leq 1$, δ_0 the point mass at 0, and U the uniform density on [0,1]. A similar analysis for the case $a \leq b$ in f_5 yields the same result.

Hence the expression in (A.1) is equivalent to

$$\arg\sup \{R^*(f) : f = \alpha\delta_0 + (1-\alpha)U\} .$$

Numerical calculation yields $\alpha^* = .382 = x^*$ and the mixture $f^* = \alpha^* \delta_0 + (1-\alpha^*)U$ is the density in F for which $R^*(f^*)$ is maximal.

Appendix 2
Justification of (4)

Let $P' = f(\mathbf{B}')$ and $P'' = f(\mathbf{B}'')$ and consider P' and P'' as discrete probability mass functions on $\Omega = \{1, \ldots, r\}$. Equip Ω with the discrete metric and let A denote a generic subset of Ω and A^c the complement (in Ω) of A. For $A \subseteq \Omega$ define $P'(A) = \sum_{i \in A} f_i(\mathbf{B}')$, $P''(A) = \sum_{i \in A} f_i(\mathbf{B}'')$, and $A^\varepsilon = \{x \in \Omega : |x - y| < \varepsilon \text{ for some } y \in A\}$. Note that $A^\varepsilon = A$ if $\varepsilon \leq 1$ and $A^\varepsilon = \Omega$, if $\varepsilon > 1$.

The Prohorov distance between P' and P'' is defined by:

$$\begin{aligned}
\rho(P', P'') &= \inf \{\varepsilon > 0 : P'(A) \leq P''(A^\varepsilon) + \varepsilon \quad \text{for all} \quad A \subseteq \Omega\} \\
&= \inf \{1 \geq \varepsilon > 0 : P'(A) - P''(A) \leq \varepsilon \quad \text{for all} \quad A \subseteq \Omega\} \\
&= \sum_{i \in C} (f_i(\mathbf{B}') - f_i(\mathbf{B}'')) \\
&= \sum_{i \in C^c} (f_i(\mathbf{B}'') - f_i(\mathbf{B}')) \\
&\equiv K
\end{aligned}$$

where $C = \{i \in \Omega : f_i(\mathbf{B}') > f_i(\mathbf{B}'')\}$.

The second to the last equality follows because

$$\sum_i f_i(\mathbf{B}') = \sum_i f_i(\mathbf{B}'') = 1 .$$

Similarly, the total variation distance between P' and P'' is given by:

$$\tau(P', P'') = \sup \{A \subseteq \Omega : |P'(A) - P''(A)| + |P'(A^c) - P''(A^c)| \} = 2K$$

Lastly, note that when $w(\mathbf{B}', \mathbf{B}'') \equiv 1$ then $\lambda(s)$ as given in (4) equals $\tau(P', P'')$ which equals $2\rho(P', P'')$.

References

ADELSON, R. M., NORMAN, J. M., and LAPORTE, G. (1976)., "A Dynamic Programming Formulation With Diverse Applications," *Operational Research Quarterly, 27*, 119-121.

ALDOUS, D. (1987), "On the Markov Chain Simulation Method for Uniform Combinatorial Distributions and Simulated Annealing," *Probability in the Engineering and Informational Sciences, 1*, 33-46.

ARABIE, P., BOORMAN, S. A., and LEVITT, P. R. (1978), "Constructing Blockmodels: How and Why," *Journal of Mathematical Psychology, 17*, 21-63.

ARABIE, P., and BOORMAN, S. A. (1982), "Blockmodels: Development and Prospects," in *Classifying Social Data: New Applications of Analytic Methods for Social Science Research*, Eds., Herschel C. Hudson and Associates, San Francisco: Jossey-Bass Publisher, Chapter 11.

BILLINGSLEY, P. (1968), *Convergence of Probability Measures*, New York: Wiley.

BOCK, H. H. (1979), "Simultaneous Clustering of Objects and Variables," in *Analyse de Données et Informatique*, Le Chesnay (France): Institut National de Recherche en Informatique et en Automatique, 187-203.

BREIGER, R. L., BOORMAN, S. A., and ARABIE, P. (1975), "An Algorithm for Clustering Relational Data With Applications to Social Network Analysis and Comparison with Multidimensional Scaling," *Journal of Mathematical Psychology, 12*, 328-383.

BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R., and STONE, C. J. (1984), *Classification and Regression Trees*, Belmont, CA: Wadsworth.

DE SOETE, G., DESARBO, W. S., FURNAS, G. W., and CARROLL, J. D. (1984), "The Estimation of Ultrametric and Path Length Trees From Rectangular Proximity Data," *Psychometrika, 49 (3)*, 289-310.

DEUTSCH, S. B., and MARTIN, J. J. (1971), "An Ordering Algorithm for Analysis of Data Arrays," *Operations Research, 19*, 1350-1362.

DIACONIS, P., and SHASHAHANI, M. (1981), "Generating a Random Permutation with Random Transpositions," *Zeitschrift fuer Wahrscheinlichkeitstheorie and verwandte Gebiete, 57*, 159-179.

DUFFY, D. E., and KEMPERMAN, J. H. B. (1990), "Entropy-Based Splitting Criteria," Morristown, NJ: Bell Communications Research Technical memorandum, in preparation.

DUFFY, D. E., FOWLKES, E. B., and KANE, L. D. (1987), "Cluster Analysis in Strategic Data Architecture Design," in 1987 *Bellcore Database Symposium*, Morristown, NJ: Bell Communications Research, Inc., 175-186.

EDELSBRUNNER, H. (1987), *Algorithms in Combinatorial Geometry*, New York: Springer-Verlag.

GILULA, Z. (1986), "Grouping and Association in Contingency Tables: An Exploratory Canonical Correlation Approach," *Journal of the American Statistical Association, 81*, 773-779.

GOODMAN, L. A. (1981), "Criteria for Determining Whether Certain Categories in a Cross-Classification Table Should be Combined With Special Reference to Occupation Categories in an Occupational Mobility Table," *American Journal of Sociology, 87*, 612-650.

GOVAERT, G. (1977), "Algorithme de Classification d'un Tableau de Contingence," *Proceedings of the First International Symposium on Data Analysis and Informatics, 2*. Le Chesnay (France): Institut National de Recherche en Informatique et Automatique, 487-500.

GREENACRE, M. J. (1988), "Clustering the Rows and Columns of a Contingency Table," *Journal of Classification, 5*, 39-51.

HARTIGAN, J. A. (1972), "Direct Clustering of a Data Matrix," *Journal of the American Statistical Association, 6*, 123-129.

HARTIGAN, J. A. (1975), *Clustering Algorithms*, New York: Wiley.

HARTIGAN, J. A. (1976), "Modal Blocks in Dentition of West Coast Mammals," *Systematic Zoology, 25*, 149-160.

HEISER, W. J., and MEULMAN, J. (1983), "Analyzing Rectangular Tables by Joint and Constrained Multidimensional Scaling," *Journal of Econometrics, 22*, 139-167.

HILL, M. O. (1974), "Correspondence Analysis: A Neglected Multivariate Method," *Applied Statistics, 23*, 340-354.

HOLLAND, P. W., and LEINHARDT, S. (1981), "An Exponential Family of Probability Distributions for Directed Graphs (with discussion)." *Journal of the American Statistical Association, 76*, 33-65.

HUBERT, L. J. (1974), "Problems of Seriation Using a Subject by Item Response Matrix," *Psychological Bulletin, 81*, 976-983.

HUBERT, L. J., and GOLLEDGE, R. G. (1981), "Matrix Reorganization and Dynamic Programming: Applications to Paired Comparisons and Unidimensional Seriation," *Psychometrika, 46*, 429-441.

LAMBERT, J. M., and WILLIAMS, W. T. (1962), "Multivariate Methods in Plant Ecology IV. Nodal Analysis," *Journal of Ecology, 50*, 775-802.

MEHTA, C. R., PATEL, N. R., and GRAY, R. (1985), "Computing an Exact Confidence Interval for the Common Odds Ratio in Several 2×2 Contingency Tables." *Journal of the American Statistical Association, 8*, 969-973.

SCHMID, B. (1984), "Niche Width and Variation Within and Between Populations in Colonizing Species (*Carex flava* group)," *Oecologia* (Berlin), *63*, 1-5.

SOKAL, R. R., and SNEATH, P. H. A. (1963), *Principles of Numerical Taxonomy*, San Francisco: Freeman.

WANG, Y. J., and WONG, G. Y. (1987), "Stochastic Blockmodels for Directed Graphs," *Journal of the American Statistical Association, 82*, 8-19.

WONG, G. Y. (1987), "Bayesian Models for Directed Graphs," *Journal of the American Statistical Association, 82*, 140-148.