Handbook of Combinatorial Optimization
D.-Z. Du and P.M. Pardalos (Eds.) pp. – – –

Combinatoral Optimization in Clustering

Boris Mirkin
Center for Discrete Mathematics & Theoretical Computer Science (DIMACS)
Rutgers University, 96 Frelinghuysen Road
Piscataway, NJ 08854-8018, USA
and Central Economics-Mathematics Institute (CEMI), Moscow, Russia.
E-mail: mirkin@dimacs.rutgers.edu

Ilya Muchnik
RUTCOR and DIMACS
Rutgers University, 96 Frelinghuysen Road
Piscataway, NJ 08854-8018, USA
E-Mail: muchnik@rutcor.rutgers.edu

Contents

1 Introduction

Clustering is a mathematical technique designed for revealing classification structures in the data collected on real-world phenomena. A *cluster* is a piece of data (usually, a subset of the objects considered, or a subset of the variables, or both) consisting of the entities which are much “alike”, in terms of the data, versus the other part of the data. The term itself was coined in psychology back in thirties when a heuristical technique was suggested for clustering psychological variables based on pair-wise coefficients of correlation. However, two more disciplines also should be credited for the outburst of clustering occurred in the sixties: numerical taxonomy in biology and pattern recognition in machine learning. Among relevant sources are Hartigan (1975), Jain and Dubes (1988), Mirkin (1996). Simultaneously, industrial and computational applications gave rise to graph partitioning problems which are touched below in 6.2.4.

Combinatorial optimization and graph theory are closely connected with clustering issues through such combinatorial concepts as connected component, clique, graph coloring, min-cut, and location problems having obvious clustering flavor. A concept interweaving the two areas of research is the minimum spanning tree (MST) emerged initially in clustering (within a biologically oriented method called Wrozlaw taxonomy, see a late reference in Florek et al. (1951)) and having become a cornerstone in computer sciences.

In the follow-up review of combinatorial clustering, we employ the most natural bases for systematization of the abundant material available: by types of input data and output cluster structures. This slightly differs of the conventional taxonomy of clustering (hierarchic versus non-hierarchic, overlapping versus non-overlapping) in which a confusion between clustering structures and algorithms may occur. In section 2, five types of data tables are considered according to extent of admitted comparability among the data entries: column-conditional, comparable, aggregable, Boolean, and spatial ones. In section 3, five types of discrete cluster structures are defined: subsets (single clusters), partitions, hierarchies, structured partitions and bipartite structures, as those the most of references deal with. A very short section 4 describes what kind of clustering criteria is the present authors' best choice, though some other criteria are also considered in the further text. A major problem with clustering criteria is that usually they cannot be clear-cut substantiated (except for those emerged in specific engineering problems): the criteria relate quite indirectly to the major goal of clustering, which is improving of our understanding of the world. This makes a great deal of clustering research to be devoted to problems of substantiation of clustering criteria with instance or Monte-Carlo studies or mathematical investigation of their properties and interconnections.

Section 5 is devoted to problems of separating a single cluster from the data (single cluster clustering). Two major ad hoc algorithms, greedy seriation and moving center separation, are discussed in the contexts of corresponding criteria and their properties. Two kinds of criteria related, monotone linkage based set functions and data approximation, are discussed at length in subsections 5.2 and 5.3, respectively. The seriation and moving center methods appear to be local search algorithms for the criteria.

Partitioning problems are considered in section 6. In subsection 6.1, the problems of partitioning for column-conditional data are discussed. The authors try to narrow down the overwhelming number of clustering criteria that have been or can be suggested. A bunch of different approaches is unified via a bunch of equivalent (under certain conditions) criteria. A bunch of ad hoc clustering methods (agglomerative clustering, K-Means, exchange, conceptual clustering) are discussed as those which appear to be local search techniques for these criteria. From the user's point of view, a major conclusion from this discussion is that the methods (along with the parameters suggested), applied to a data set, will yield similar results. The optimal partitioning problem in the coordinate-based framework seems under-studied and needed more efforts. In subsection 6.2, partitioning of (dis)similarity (comparable) data matrices is covered. The topics of interest are: uniform partitioning, additive partitioning, and graph partitioning discussed mostly in the context of data approximation. The last part is devoted to the problem of structured partitioning (block modeling). In subsection 6.3, the approximation approach is applied to clustering problems with no nonoverlapping restrictions imposed.

Hierarchies as clustering structures are discussed in section 7. In subsection 7.1, an approximation model is shown to lead to some known ad hoc divisive clustering techniques. The other subsections deal with indexed hierarchies (ultrametrics) and tree metrics, the subject of particular interest in molecular evolution studies (Setubal and Meidanis (1997)).

Section 8 is devoted to three approximation clustering problems for aggregable (co-occurrence) data: box clustering (revealing associated row-column sets), bipartitioning/aggregation of rectangular matrices, and aggregation of square interaction (flow) matrices. The aggregable data seem of importance in a predictable future since they present information about very large or massive data sets in a treatable format of counts or volumes.

The material is illustrated with examples which are printed with a smaller font.

For additional coverage, see Brucker (1978), Arabie and Hubert (1992), Crescenzi and Kann (1995), Arabie, Hubert and De Soete [Eds.] (1996), Day (1996) and Mirkin (1996).

2 Types of Data

Mathematical formulations for clustering problems much depend on the accepted format of input data. Though in the real world more and more data are of continuous nature, as images and signals, the computationally treated cases involve usually discrete or digitalized data. The discrete data are considered usually as arranged in a table format.

To get an intuition on that, let us consider a set of data presented in Table 1 which is just a 7×7 matrix, $X = (x_{ik})$, $i \in I$, $k \in K$. Three features of the table are due to the authors' willingness of using the same data set for illustrating many problems. In general, the entries may be any reals, not just zeros and ones; there may be no symmetry in the matrix entries, and the number of rows may differ from the number of columns. Data in tables 6 and 7 are instances of such more general data sets.

Table 1: An illustrative data set (the zero entries are not shown).

Columns	1	2	3	4	5	6	7
Rows							
1	1	1	1				
2	1	1	1		1		
3	1	1				1	
4					1	1	1
5		1		1		1	1
6			1	1	1	1	1
7				1	1	1	1

Depending on the extent of comparability among the data entries, it is useful to distinguish among the following data types:

(A) Column-Conditional Data.

(B) Comparable Data.

(C) Aggregable Data.

(D) Boolean Data.

(E) Spatial Data.

The meaning of these follows.

(A) Column-Conditional Data

The columns are considered different noncomparable variables so that their entries may be compared only within the columns. For instance, suppose every row is a record of the values of some variables for an individual, so that the first column of X relates to sex (0 - male, 1 - female) while the second to the respondent's attitude toward a particular kind of cereal (1 - liking, 0 - not liking).

In such a situation, a preliminary transformation is usually performed to standardize the columns so that they could be thought of as comparable, to some extent. Such a standardizing transformation usually is

$$y_{ik} := \frac{x_{ik} - a_k}{b_k}, \tag{2.1}$$

to shift the origin (to a_k) and change the scale (by factor b_k) where a_k is a central or normal point in the range of the variable (column) k and b_k a measure of the variable's dispersion. When a hypothesis about a probabilistic distribution as the variable's generating facility can be admitted with no much violation of the data's nature, the standardizing parameters can be taken from the distribution theory, as the average, for a_k, and standard deviation, for b_k, when the distribution is Gaussian. When no reliable and reproducible mechanism for the data generation can be assumed, the choice of the parameters should be based on a different way of thinking as, say approximation considerations in Mirkin (1996). The least-squares approximation also leads to the average and standard deviation as the most appropriate values. The standardized matrix $Y = (y_{ik})$ obtained with these shift and scale parameters is presented in Table 2 which is not symmetric anymore. However, other approximation criteria may lead to differently defined a_k and b_k. For instance, the least-maximum-deviation criterion yields b_k as the range and a_k mid-range of the variable k.

(B) Comparable Data

A data table $X = (x_{ik})$ is comparable if all the values x_{ik} ($i \in I$, $k \in K$; sometimes $I = K$) across the table are comparable, which means also that the user considers it is convenient to average any subset of the entries.

The original data in Table 1 can be considered comparable if they present, say, an account of mutual liking among seven individuals. Also, comparable data tables are frequently obtained from the column-conditional tables as between-item similarities or dissimilarities. Similarity differs from dissimilarity by direction: increase in difference between two items corresponds to a smaller similarity and larger dissimilarity value. A dissimilarity table is

Table 2: Matrix Y obtained from X via least-squares standardization.

1	1.15	0.87	1.15	-0.87	-1.15	-1.58	-1.15
2	1.15	0.87	1.15	-0.87	0.87	-1.58	-1.15
3	1.15	0.87	-0.87	-0.87	-1.15	0.63	-1.15
4	-0.87	-1.15	-0.87	-0.87	0.87	0.63	0.87
5	-0.87	0.87	-0.87	1.15	-1.15	0.63	0.87
6	-0.87	-1.15	1.15	1.15	0.87	0.63	0.87
7	-0.87	-1.15	-0.87	1.15	0.87	0.63	0.87

called distance if it satisfies the metric space axioms (more on dissimilarities see in [76], [61]). A graph with its edges weighted can be considered as a nonnegative comparable $|I| \times |I|$ similarity matrix (of the weights).

Tables 3 and 4 present similarity matrices obtained from Table 1 considered as a column-conditional table. Table 3 is a distance matrix. Its (i,j)-th entry h_{ij} is the number of noncoinciding components in the row-vectors, which is called Hamming distance. The other preferred distances are Euclidean distance squared,

$$d^2_{ij} := \sum_{k \in J} |x_{ik} - x_{jk}|^2,$$

and the city-block metric,

$$d_c := \sum_{k \in J} |x_{ik} - x_{jk}|.$$

Curiously, because of binary entries, these latter distances coincide, in this particular case, with each other and with the Hamming distance.

Table 4 is matrix $A = YY^T$ of scalar products of the rows of matrix Y in Table 2. It is a similarity matrix.

There exists an evident connection between the Euclidean distance squared and the scalar product similarity measure derived from the same entity-to-variable table:

$$d^2_{ij} = (y_i, y_i) + (y_j, y_j) - 2(y_i, y_j) \qquad (2.2)$$

where $a_{ij} := (y_i, y_j) := \sum_{k \in K} y_{ik} y_{jk}$, which allows for converting the scalar product similarity matrix $A = YY^T$ into the distance matrix $D = (d_{ij})$

Table 3: Matrix H of Hamming distances between the rows of X.

Entity	1	2	3	4	5	6	7
1	0	1	2	6	5	6	7
2	1	0	3	5	6	5	6
3	2	3	0	4	3	6	5
4	6	5	4	0	3	2	1
5	5	6	3	3	0	3	2
6	6	5	6	2	3	0	1
7	7	6	5	1	2	1	0

Table 4: Similarity matrix $S = YY$.

Entity	1	2	3	4	5	6	7
1	1.33	1.00	0.50	-0.75	-0.42	-0.67	-1.00
2	1.00	1.25	0.17	-0.50	-0.75	-0.42	-0.75
3	0.50	0.17	0.95	-0.30	0.03	-0.80	-0.55
4	-0.75	-0.50	-0.30	0.78	-0.05	0.28	0.53
5	-0.42	-0.75	0.03	-0.05	0.87	0.03	0.28
6	-0.67	-0.42	-0.80	0.28	0.03	0.95	0.62
7	-1.00	-0.75	-0.55	0.53	0.28	0.62	0.87

rather easily: $d^2_{ij} = a_{ii} + a_{jj} - 2a_{ij}$. The reverse transformation, converting the distances into the scalar products, can be defined when all columns in Y are centered, which means that the sum of all the row-vectors is equal to the zero vector, $\sum_{i \in I} y_i = 0$. In this case,

$$(y_i, y_j) = -\frac{1}{2}(d^2_{ij} - d^2_{i+} - d^2_{+j} + d^2_{++}) \qquad (2.3)$$

where d^2_{i+}, d^2_{+j}, and d^2_{++} denote the within-row mean, within-column mean, and the grand mean, respectively, in the array (d^2_{ij}).

Frequently, the diagonal entries (a.k.a. (dis)similarities of the entities with themselves) are of no interest or just unmeasured; this does not much affect the problems and algorithms; in the remainder, the diagonal entries present will be the default option.

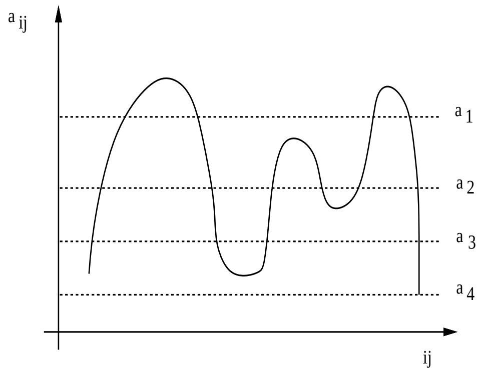

Figure 1: The effect of shifting the origin of a similarity measure.

For standardizing a dissimilarity matrix, there is no need to change the scale factor since all the entries are comparable across the table. On the other hand, shifting the origin by subtracting a threshold value a, $b_{ij} := a_{ij} - a$ where b_{ij} is the index shifted, may allow a better manifestation of the structure of the data. Fig. 1 illustrates how the shifting affects the shape of a similarity index, a_{ij}, whose values are the ordinates while ijs are put somehow on the abscissa: shifting by a_4 does not make much difference since all the similarities remain positive; shifting by a_2, a_3, or a_1 makes many similarities negative leaving just a couple of the higher similarity "islands" positive. We can see that an intermediate $a = a_2$ manifests all the three humps on the picture, while increasing it to a_1 loses some (or all) of the islands in the negative depth.

Quite a clustering structure is seen in Table 4 (the mean of which is obviously zero): its positive entries correspond to almost all similarities within two groups, one consisting of the entities 1, 2, and 3 and the other of the rest.

(C) Aggregable Data

When the data entries measure or count the number of occurrences (as in contingency tables) or volume of some matter (money, liquid, etc.) so that all of them can be summed up to the total value, the data table is referred to as the aggregable (summable) one. In such a table the row or/and column items can be aggregated, according to their meaning, in such a way that the corresponding rows and columns are just summed together.

Example. Considering Table 1 as a data set on patterns of phone calls made by the row-individuals to the column-individuals and aggregating the rows in $V_1 =$

$\{1,2,3\}$, $V_2 = \{4,5\}$, $V_3 = \{6,7\}$, and columns in $W_1 = \{1,3,5\}$ and $W_2 = \{2,4,6,7\}$, we get the aggregate phone call chart on the group level in table 5.

Table 5: Table X aggregated.

	T_1	T_2
S_1	6	4
S_2	1	6
S_3	3	6

□

Example. A somewhat more realistic data set is presented in table 6 reporting results of a psychophysical experiment on confusion between segmented numerals (Keren and Baggen (1981)).

Table 6: **Confusion:** Keren and Baggen (1981) data on confusion of the segmented numeral digits 0 to 9.

	Response									
Stimulus	1	2	3	4	5	6	7	8	9	0
1	877	7	7	22	4	15	60	0	4	4
2	14	782	47	4	36	47	14	29	7	18
3	29	29	681	7	18	0	40	29	152	15
4	149	22	4	732	4	11	30	7	41	0
5	14	26	43	14	669	79	7	7	126	14
6	25	14	7	11	97	633	4	155	11	43
7	269	4	21	21	7	0	667	0	4	7
8	11	28	28	18	18	70	11	577	67	172
9	25	29	111	46	82	11	21	82	550	43
0	18	4	7	11	7	18	25	71	21	818

□

Example. Yet another, rectangular, contingency data matrix is in table 7 (from L. Guttman, 1971 as presented in Mirkin (1996)) which cross-tabulates 1554 Israeli adults according to their living places as well as, in some cases, those of their fathers (column items) and "principal worries" (row items). There are 5 column items considered: EUAM - living in Europe or America, IFEA - living in Israel, father living in Europe or America, ASAF - living in Asia or Africa, IFAA- living in Israel, father living in Asia or Africa, IFI - living in Israel, father also living in Israel. The principal worries are: POL, MIL, ECO - political, military and economical situation, respectively; ENR - enlisted relative, SAB - sabotage, MTO - more than one worry, PER - personal economics, OTH - other worries.

Table 7: **Worries:** The original data on cross-classification of 1554 individuals by their worries and origin places.

	EUAM	IFEA	ASAF	IFAA	IFI
POL	118	28	32	6	7
MIL	218	28	97	12	14
ECO	11	2	4	1	1
ENR	104	22	61	8	5
SAB	117	24	70	9	7
MTO	42	6	20	2	0
PER	48	16	104	14	9
OTH	128	52	81	14	12

□

This kind of data traditionally is not distinguished from the others, which makes us to discuss it in more detail. Let us consider an aggregable data table $P = (p_{ij})$ $(i \in I, j \in J)$ where $\sum_{i\in I}\sum_{j\in J} p_{ij} = 1$, which means that all the entries have been divided by the total $p_{++} = \sum p_{ij}$. Since the matrix is non-negative, this allows us to treat p_{ij}s as frequencies or probabilities of simultaneously occurring row $i \in I$ and column $j \in J$ (though, no probabilistic estimation problems will be considered in this chapter). Note that the rows and columns of such a table are usually some categories.

The only transformation we suggest for the aggregable data is

$$q_{ij} = \frac{p_{ij}}{p_{i+}p_{+j}} - 1 = \frac{p_{ij} - p_{i+}p_{+j}}{p_{i+}p_{+j}} \tag{2.4}$$

where $p_{i+} = \sum_{j\in J} p_{ij}$ and $p_{+j} = \sum_{i\in I} p_{ij}$ are the so-called marginals equal to the totals in corresponding rows and columns.

When the interpretation of p_{ij} as co-occurrence frequencies is maintained, q_{ij} means the relative change of probability (RCP) of i when column j becomes known, RCP(i/j)=(p(i/j)-p(i))/p(j). Here, $p(i/j) := p_{ij}/p_{+j}$, $p(i) = p_{i+}$, and $p(j) = p_{+j}$. Symmetrically, it can be interpreted also as RCP(j/i). The ratio, $\frac{p_{ij}}{p_{i+}p_{+j}}$, is frequently referred to as the odds-ratio. In the general setting, p_{ij} may be considered as amount of flow, or transaction from $i \in I$ to $j \in J$. In this case, $p_{++} = \sum_{i,j} p_{ij}$ is the total flow, $p(j/i)$ defined as $p(j/i) = p_{ij}/p_{i+}$, the share of j in the total transactions of i, and

$p(j) = p_{+j}/p_{++}$ is the share of j in the overall transactions. This means that the ratio $p(j/i)/p(j) = p_{ij}p_{++}/(p_{i+}p_{+j})$ compares the share of j in i's transactions with the share of j in the overall transactions. Then,

$$q_{ij} = p(j/i)/p(j) - 1$$

shows the difference of transaction p_{ij} with regard to "general" behavior: $q_{ij} = 0$ means that there is no difference in $p(j/i)$ and $p(j)$; $q_{ij} > 0$ means that i favors j in its transactions while $q_{ij} < 0$ shows that the level of transactions from i to j is less than it is "in general"; value q_{ij} expresses the extent of the difference and can be called *flow index*. Equation $q_{ij} = 0$ is equivalent to $p_{ij} = p_{i+}p_{+j}$ which means that row i and column j are *statistically independent* (under the probabilistic interpretation). In the data analysis context, $q_{ij} = 0$ means that knowledge of j adds nothing to our ability in predicting i, or, in the flow terms, that there is no difference between the pattern of transactions of i to j and the general pattern of transactions to j.

The smaller p_{i+} and/or p_{+j}, the larger q_{ij} grows. For instance, when p_{i+} and p_{+j} are some 10^{-6}, q_{ij} may jump to million while the other entries will be just around unity. This shows that the transformation (2.4), along with the analyses based on that, should not be applied when the marginal probabilities are too different.

Example. The table $Q = (q_{ij})$ for the Worries set is in Table 8.

Table 8: Values of the relative changes of probability (RCP), multiplied by 1000, for the Worries data.

	EUAM	IFEA	ASAF	IFAA	IFI
POL	222	280	-445	-260	36
MIL	168	-338	-129	-234	72
ECO	145	-81	-302	239	487
ENR	28	-40	11	-58	-294
SAB	19	-77	22	-66	-129
MTO	186	-252	-53	-327	-1000
PER	-503	-269	804	726	331
OTH	-118	582	-65	149	181

□

Taking into account the summability of the data (to unity), the distance between the row (or column) entities should be defined by weighting the columns (or rows) with their "masses" p_{+j} (or, respectively, p_{i+}), as for instance,

$$\chi^2(i,i') = \sum_{j\in J} p_{+j}(q_{ij} - q_{i'j})^2. \tag{2.5}$$

This is equal to the so-called chi-squared distance considered in the theory of a major visualization technique, the correspondence analysis (see, for example, Benzécri (1973) and Lebart, Morineau and Piron (1995)), and defined, in that theory, with the *profiles* of the conditional probability vectors $y_i = (p_{ij}/p_{i+})$ and $y_{i'} = (p_{i'j}/p_{i'+})$:

$$\chi^2(i,i') := \sum_{j\in J}(y_i - y_{i'})^2/p_{+j} = \sum_{j\in J}(p_{ij}/p_i - p_{i'j}/p_{i'})^2/p_{+j}.$$

(4) Boolean Data

Boolean (yes/no or one/zero) data are supposed to give, basically, set-theoretic information. Due to such a table $X = (x_{ij})$, any row $i \in I$ is associated with the set W_i of columns j for which $x_{ij} = 1$ while any column $j \in J$ is associated with the row set V_j consisting of those i for which $x_{ij} = 1$. Supposedly there is no other information in the table beyond that. This is usually presented in the graph-theoretic format to allow all the graph theory constructions applicable. Considering Table 1 as a Boolean similarity table, it corresponds to the graph presented in Fig. 2.

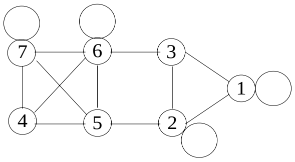

Figure 2: Graph corresponding to Table 1.

However due to its binariness this kind of data can be treated also as any other type considered above, especially as comparable or aggregable data.

In VLSI or parallel computing applications, the entities are nodes of a two- or more-dimensional greed (mesh) which is frequently irregular. This coordinate-based information can be translated into a sparse graph format in the following way (see Miller, Teng, Thurston, and Vavasis (1993)). A k-ply neighborhood system for a data matrix Y is defined as a set of closed balls in R^n, such that no point y_i, $i \in I$, is strictly interior to more than k balls. An (α, k) overlap graph is a graph defined in terms of a constant

$\alpha \geq 1$, and a k-ply neighborhood system $\{B_1, ..., B_q\}$. There are q nodes, each corresponding to a ball, B_m. There is an edge (m, l) in the graph if expanding the radius of the smaller of B_m and B_l by a factor α causes the two balls to overlap.

(E) Spatial Data

These are the tables reflecting the plane continuity of a two-dimensional space so that the rows and columns represent sequential strips of the plane, and the entries correspond to observations in their intersection zones. A typical example: any (2-D) digitalized image presented via brightness value at every pixel (cell) of a roster (greed). For instance, the data in Table 1 can be thought of as an 8×8 greed with the unities standing for darker cells. Formally, the spatiality is reflected in the fact that both rows and columns are totally ordered according to the greed so that comparing two cells should involve all the intermediates.

We do not consider here what is called *multiway* tables related to more than two index sets of the data tables (as, for instance, 3-D images or the same table measurements made in different locations/times).

3 Cluster Structures

The following categories of combinatorial cluster structures to be revealed in the data can be found in the literature:

1. *Subsets (Single Clusters).* A subset $S \subseteq I$ is a simple classification structure which can be represented in any of three major forms:

   a) Enumeration, $S = \{i_1, i_2, ..., i_m\}$;

   b) Boolean indicator vector, $s = (s_i)$, $i \in I$, where $s_i = 1$ if $i \in I$ and $s_i = 0$ otherwise;

   c) Intensional predicate, $P(i)$, defined for all $i \in I$, which is true if and only if $i \in S$.

   The latter format can be considered as belonging in the class of "conceptual structures".

2. *Partitions.* A set of nonempty subsets S=$\{S_1, ..., S_m\}$ is called a partition if and only if every element $i \in I$ belongs to one and only one of these subsets called classes; that is, S is a partition when $\cup_{t=1}^{m} S_t = I$, and $S_t \cap S_u = \emptyset$ for $t \neq u$.

3. *Hierarchies.* A hierarchy is a set $S_W = \{S_w : w \in W\}$ of subsets $S_w \subseteq I$, $w \in W$ (where W is an index set), called *clusters* and satisfying the following conditions: 1) $I \in S_W$; 2) for any $S_1, S_2 \in S_W$, either they are nonoverlapping ($S_1 \cap S_2 = \emptyset$) or one of them includes the other ($S_1 \subseteq S_2$ or $S_2 \subseteq S_1$), which can be expressed as $S_1 \cap S_2 \in \{\emptyset, S_1, S_2\}$. Throughout this chapter, yet one more condition will be assumed: (3) for each $i \in I$, the corresponding singleton is a cluster, $\{i\} \in S_W$. This latter condition guarantees that any non-terminal cluster is union of the singletons it contains. Such a hierarchy can be represented graphically by a rooted tree: its nodes correspond to the clusters (the root, to I itself), and its leaves (also called terminal or pendant nodes), to the minimal clusters of the hierarchy, which is reflected in the corresponding labeling of the leaves. Since this picture very much resembles that of a genealogy tree, the immediate subordinates of a cluster are called its children while the cluster itself is referred to as their parent.

4. *Structured Partition (Block Model).* A structured partition is a partition $S = \{S_t\}$, $t \in T$, on I, for which a supplementary relation (graph) $\omega \subset T \times T$ is given to represent "close" association between corresponding subsets S_t, S_u when $(t, u) \in \omega$ (so that (S, ω) is a "small" graph).

5. *Bipartite Clustering Structures.* This concept is defined when the data index sets, I and J (or K), are considered as different ones. The following bipartite clustering structures involve single subsets, partitions, and hierarchies to interconnect I and J: 1) box (V, W), $V \subseteq I$, $W \subseteq J$; 2) bipartition, a pair of partitions, (S, T), with S defined on I and T on J, along with a correspondence between the classes of S and T; 3) bihierarchy, a pair of interconnected hierarchies, (S_F, T_H), $S_F \subset 2^I$, $T_H \subset 2^J$.

4 Clustering Criteria

When a data set is given and a type of clustering structure has been chosen, one needs a criterion to estimate degree of fit between the structure and data. Initially, it was a lot of ad hoc criteria suggested in clustering (see, for instance, Brucker (1978) and Arabie, Hubert and De Soete [Eds.] (1996)). Currently, the following way of thinking seems more productive.

To measure goodness-of-fit, the cluster structure sought, A, should be employed to formally reconstruct the data matrix, $X(A)$, as if it would have been produced by the structure A only, with no noise and other influences interfered. In this case, relation between the original data matrix, X, and the cluster structure, A, can be stated as the following equation:

$$X = X(A) + E \tag{4.1}$$

where E stands for the matrix of residuals, $E = X - X(A)$, which should be minimized with regard to the admissible cluster structures A.

Though equation (4.1) may be considered as appealing to statistics framework, no statistical model for the residuals has been developed so far in such a setting. The operations research multigoal perspective (compromise minimizing of all residuals simultaneously) also seems foreign to clustering. The only framework being widely developed is approximation clustering in which the clustering problems are considered as those of minimizing a norm of E with regard to admissible cluster structures. Three norms are in use currently: L^2, the sum of the residual entries squared, L_1, the sum of absolute values of the residuals, and L_∞, the maximum absolute value in E. The problem of minimizing of one of these criteria is referred to as the least-squares, least-moduli, and the least-deviation method, respectively.

In the remainder, we will describe clustering problems according to the type of cluster structure to reveal. When the data and cluster structure types are chosen, a criterion of fit may be defined based on substantive or heuristical considerations, which will be also considered when appropriate.

5 Single Cluster Clustering

5.1 Clustering Approaches

There are three major approaches to determine a cluster as based on: a) Definition, (b) Direct algorithm, and (c) Optimality criterion.

5.1.1 Definition-based Clusters

A cluster is thought of as a subset $S \subseteq I$ consisting of very "similar" entities. Its dual, an "anti-cluster", is to consist of mutually remote entities.

Let B_i be a subset of entities which are "similar" to $i \in I$. Such a subset can be defined as the set of adjacent vertices in a graph connecting entities or as a “ball” of entities whose (dis)similarity to i is (not) greater than a threshold.

A subset $S \subseteq I$ can be referred to as a component cluster if, for any $i \in S$, it contains B_i, and as a clique cluster if, for any $i \in S$, it is contained in B_i and no larger set satisfies this property. The component and clique clusters are components and cliques, respectively, in the graph (I, B) defined by the adjacency subsets B_i. Anti-cluster concepts involve independent subsets in graphs.

Some more cluster concepts in terms of dissimilarities:

(1) A *clump* cluster is such a subset S that, for every $i, j \in S$ and $k, l \in I - S$, $d_{ij} < d_{kl}$. Obviously, any clump is a clique and connected component in a threshold graph (I, B) defined by the condition $(i, j) \in B$ iff $d_{ij} < \pi$ where threshold π is taken between $\max_{i,j \in S} d_{ij}$ and $\min_{k,l \in I-S} d_{kl}$.

(2) A *strong* cluster is such a subset S that, for every $i, j \in S$ and $k \in I - S$, $d_{ij} < d_{ik}$, or, which is the same, $d_{ij} < \min(d_{ik}, d_{jk})$. Any strong cluster is simultaneously a clique and connected component in graph (I, B) whose adjacency sets B_i are defined by condition $d_{ij} < \pi_i$ where threshold π_i is taken between $\max_{j \in S} d_{ij}$ and $\min_{k \in I-S} d_{ik}$. If two strong clusters overlap, then one of them is a part of the other. They form a (strong) hierarchy: $S_1 \cap S_2 \in \{\emptyset, S_1, S_2\}$, for any strong clusters S_1, S_2. I may be considered a strong cluster as well. Obviously the clump clusters also fit in these properties.

(3) A *weak* cluster is defined by a weak form of the condition above: $d_{ij} < \max(d_{ik}, d_{jk})$ for all $i, j \in S$ and $k \notin S$. Weak clusters form a weak hierarchy: $S_1 \cap S_2 \cap S_3 \in \{\emptyset, S_1 \cap S_2, S_2 \cap S_3, S_3 \cap S_1\}$, for any weak clusters S_1, S_2, S_3 (Bandelt and Dress 1989).

(4) A π-cluster $S \subseteq I$ is defined by condition that $d(S) \leq \pi$ where $d(S) = \sum_{i,j \in S} d_{ij}/|S||S|$ is the average dissimilarity within S.

(5) A *strict* cluster is such an $S \subseteq I$ that, for any $k \in I - S$ and $l \in S$, $d(l, S) \leq 2d(S) < d(k, S)$ where $d(i, S)$ is the average dissimilarity of i and S.

All these concepts are trivially redefined in terms of similarities except for the strict clusters whose defining conditions become: for any $k \in I - S$ and $l \in S$, $a(l, S) \leq a(S)/2 < a(k, S)$. (Here $a(S)$ and $a(k, S)$ are the average similarities.)

Finding clumps and component clusters involves finding cliques and components in graphs; the other concepts are not as well developed.

5.1.2 Direct Algorithms

In clustering, it is not uncommon to use a cluster designing technique with no explicit model behind it at all: such a technique itself can be considered a model of clustering process. Two of such direct clustering techniques are seriation and moving center, both imitating some processes in typology making.

A seriation procedure involves a (dis)similarity linkage measure $d(i,S)$ evaluating degree of dissimilarity between subsets $S \subset I$ and entities $i \in I - S$.

Seriation
Initial setting: $S = \emptyset$ if $d(i,\emptyset)$ is defined for all $i \in I$, or, otherwise, $S = \{i_0\}$, $i_0 \in I$ being an arbitrary entity.
General step: given S, find $i^* \in I - S$ minimizing dissimilarity linkage measure $d(i,S)$ with regard to all $i \in I - S$ and join i^* as the last element in S seriated.

The general step is repeated until a stop-condition is satisfied. It is also possible that the final cluster(s) is cut out of the ordering of entire set I resulting from the seriation process.

The seriation, actually, is a greedy procedure.

Examples of linkage functions:

A For dissimilarity matrices, $D = (d_{ij})$:

1. *Single linkage* or *Nearest neighbor*

$$sl(i,S) = \min_{j \in S} d_{ij};$$

2. *Summary linkage*

$$su(i,S) = \sum_{j \in S} d_{ij};$$

B For similarity matrices, $A = (a_{ij})$:

3. *Average linkage* or *Average neighbor*

$$al(i,S) = \sum_{j \in S} a_{ij}/|S|;$$

4. *Threshold linkage*

$$l_\pi(i,S) = \sum_{j \in S}(a_{ij} - \pi) = \sum_{j \in S} a_{ij} - \pi|S|;$$

where π is a fixed threshold value.

C For column-conditional matrices, $Y = (y_{ik})$:

5. *Entity-to-center scalar product*

$$a(i, S) = (y_i, c(S))$$

where $y_i = (y_{ik})$, $k \in K$, is i-th row vector and $c(S)$ is the gravity center of S, $c(S) = \sum_{j \in S} y_j/|S|$, and (x, y) stands for the scalar product of vectors x and y.

This measure obviously coincides with the average linkage (3.) when the similarity matrix is defined as $a_{ij} = (y_i, y_j)$.

6. *Holistic linkage*

$$hl(i, S) = \sum_{k \in K} \min_{j \in S} |y_{ik} - y_{jk}|$$

D For spatial data arrays:

7. *Window linkage*

$$d_W(i, S) = d(i, S \cap W_i)$$

where W_i is a window around i in the data array (usually, window W_i is an $m \times m$ cell square put on the data greed so that i is in the window center).

A natural stopping rule in the seriation process can be used when the threshold linkage is employed: i^* is not added to S if $l_\pi(i^*, S) < 0$. It is not difficult to prove that the result is a π-cluster.

Another single cluster clustering techniques separates a cluster from the "main body" of the entities.

Moving Center

Initial setting: a tentative center of the cluster to form, c, and a constant center of the main body (a "reference point"), a. Usually the origin is considered shifted so that $a = 0$.

General procedure iterates the following two steps:

1) (Updating of the cluster) define the cluster as the ball of radius $R(a,i)$ around c: $S = \{i : d(i,c) \leq R(a,i)\}$. Two most common definitions: (a) constant radius, $R(a,i) = r$, where r is a constant, (b) distance from the reference point, $R(a,i) = d(a,i)$.

2) (Updating the center) define $c = c(S)$, a centroid of S.

The process stops when the newly found cluster S coincides with that found at the previous iteration.

The algorithm may or may not converge (see subsection 5.3 below).

Curiously, in the reference-point-based version of the algorithm, the cluster size depends on the distance between c and a; the less the distance, the less the cluster radius. This feature can be useful, for example, when a moving robotic device classifies the elements of its environment: the greater the distance, the greater the clusters, since differentiation among the nearest matters more for the robot's moving and acting.

5.1.3 Optimal Clusters

Clustering criteria may come up from particular problems in engineering or other applied area or from clustering considerations.

Two most known "engineering" clustering problems are those of knapsack and location.

The knapsack problem is of finding a subset S maximizing its weight, $\sum_{i \in S} w_i$, while keeping other parameter(s) constrained.

This problem is NP-complete though admits a polynomial time approximation (see Garey and Johnson (1979) and Crescenzi and Kann (1995)).

The location problem is to find a subset S minimizing the cost

$$f(S) = \sum_{i \in S} f_i + \sum_j \min_{i \in S} c_{ij}$$

where i is a location, f_i its cost, and c_{ij} the cost of transport of the product from warehouse i to customer j.

When $C = (c_{ij})$, $i,j \in I$, is non-negative, the function f is supermodular; that is, it satisfies inequality

$$f(S_1 \cup S_2) + f(S_1 \cap S_2) \geq f(S_1) + f_(S_2)$$

for any $S_1, S_2 \subseteq I$. This is also a hard problem (see Hsu and Nemhauser (1979) and Gondran and Minoux (1984), p. 461).

Among the problems formalizing single cluster clustering goals as they are, two most popular are those of maximum clique and maximum density subgraph.

The problem of finding a maximum size clique in a graph belongs to the core of combinatorial optimization. It is NP-complete, but admits some approximations (see the latest news in Johnson and Trick [Eds.] (1996) and Crescenzi and Kann (1995)).

The maximum density subgraph problem is to find a subset S maximizing the ratio

$$g(S) = \frac{\sum_{i,j \in S} a_{ij}}{|S|}$$

of the total weight of edges within S to the number of vertices in S. (Here $A = (a_{ij})$ is the edge weight matrix.) The problem can be reduced to a restricted number of solutions to the problem of maximizing a linearized version of g, $G(S, \lambda) = \sum_{i,j \in S} a_{ij} - \lambda |S|$. Function $G(S, \lambda)$ is a supermodular function so that the problem can be solved in a polynomial time (see Gallo, Grigoriadis, and Tarjan (1989) where a max-flow technique is exploited for the problem).

The function of maximum density obviously has something to do with the average linkage function, $g(S) = \sum_{i \in S} al(i, S)$. This illustrates that many single cluster clustering criteria can be obtained by integrating of linkage (dis)similarity functions with regard to their argument subsets: $D_d(S) = INT_{i \in S} d(i, S)$ where INT can be any operation with reals, as summation or averaging or taking minimum.

In the remainder of this section, two topics related to the direct clustering techniques will be treated in more detail: (a) single linkage clustering and its extensions (subsection 5.2), and (b) models underlying the moving center algorithm (subsection 5.3).

5.2 Single and Monotone Linkage Clusters

5.2.1 MST and Single Linkage Clustering

Let $D = (d_{ij})$ be a symmetric $N \times N$ matrix of the dissimilarities d_{ij} between elements $i, j \in I$.

The concept of minimum spanning tree (MST) is one of the most known in combinatorial optimization. A spanning tree, $T = (I, V)$, with I the set of its vertices, is said to be an MST if its length, $d(T) = \sum_{(i,j)\in V} d_{ij}$, is minimum over all possible spanning trees on I. Two of the approaches to finding an MST are to be mentioned here. In one of the approaches, Kruskal algorithm, an MST is produced by starting with an empty edge set, $V = \emptyset$, and greedily adding edge by edge, (i,j) to V, in the order of increasing d_{ij} (maintaining V with no cycles). The other approach, Prim algorithm, works differently. It processes vertices, one at a time, starting with $S = \emptyset$ and updating S at each step by adding to S an element $i \in I - S$ minimizing its single linkage distance to S, $sl(i,S) = \min_{j\in S} d_{ij}$.

The former approach has been generalized into a theory of greedy optimization of linear set functions based on the matroid theory (Welsh (1976)). The set of all edge subsets with no cycles is a matroid; greedily adding an edge by edge (Kruskal algorithm) produces an MST.

As to the latter approach, its important properties can be stated as these (Delattre and Hansen (1980)): Let us define the so-called minimum split set function, $L(S) = \min_{j\in I-S} \min_{i\in S} d_{ij}$, as a measure of dissimilarity between S and $T - S$. Let us refer to an $S \subset I$ as a minimum split cluster if S is a maximizer of $L(S)$ over the set $\mathcal{P}^{-}(I)$ of all non-empty proper sets $S \subset I$. All inclusion-minimal minimum split clusters can be found by cutting any MST at all its maximum weight edges. Obviously, all unions of these minimal clusters are also minimum split clusters.

There have been no attempts made to generalize this approach until recently. Note that the seriation algorithm above is a natural extension of the Prim algorithm. Based on the single linkage dissimilarity function, $sl(i,S) := \min_{j\in S} d_{ij}$, the seriation algorithm defines a *single linkage series*, $s = (i_1, i_2, ..., i_N)$, by the condition that for every $k = 1, ..., N-1$, the element i_{k+1} is a minimizer of $sl(i, S_k)$ with regard to $i \in I - S_k$. Here $S_k := \{i_1, ..., i_k\}$ is a starting set of the series s ($k = 1, 2, ..., N-1$). The Prim algorithm finds a minimum linkage series s and an MST associated with it: its edges connect, for $k = 1, ..., N-1$, the vertex i_{k+1} with just one of the vertices $j \in S_k$ that have $d_{i_{k+1}j} = sl(i_{k+1}, S_k)$. The minimum split clusters are what can be called single linkage clusters, that is, starting sets S_k of single linkage series s, which are maximally separated from the other elements along the series. More explicitly, a single linkage cluster is S_k in a single linkage series $s = (i_1, i_2, ..., i_N)$ such that $sl(i_{k+1}, S_k)$ is maximum over all $k = 1, ..., N-1$ (which is a "greedy" definition).

Example. The only MST for the matrix in table 3 of Hamming distances between rows of the Boolean matrix in table 1 is presented in Fig. 3. Similarity matrix in table 4 implies the same tree (with different edge weights) as the maximum

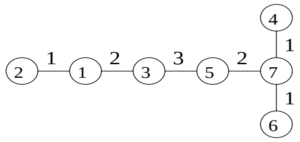

Figure 3: Minimum spanning tree for the distance data in table 3.

spanning tree. There is only one maximum length edge $(3,5)$; by cutting it out, we obtain two inclusion-minimal minimum split clusters, $\{1,2,3\}$ and $\{4,5,6,7\}$. □

In the next subsection, this construction will be extended to the problem of greedy optimization of yet another class of set functions, quasi-concave, not linear ones, as described in Kempner, Mirkin and Muchnik (1997).

5.2.2 Monotone Linkage Clusters

A version of the greedy seriation algorithm finds minimum split clusters for a class of minimum split functions defined with the so-called monotone linkage functions. Let us refer to a linkage function $d(i,S)$, $S \in \mathcal{P}^-(I)$, $i \in I - S$, as a *monotone linkage* if $d(i,S) \geq d(i,T)$ whenever $S \subseteq T$ (for all $i \in I - T$). Given a linkage function d, a set function M_d called the *minimum split function* for d is defined by

$$M_d(S) := \min_{i \in I-S} d(i,S). \tag{5.1}$$

It appears, that the set of minimum split functions of the monotone linkages coincides with the set of $\cap$-concave set functions (Kempner, Mirkin and Muchnik (1997)). A set function $F : \mathcal{P}^-(I) \rightarrow R$ will be referred to as $\cap$-*concave* if

$$F(S_1 \cap S_2) \geq \min(F(S_1), F(S_2)), \tag{5.2}$$

for any overlapping $S_1, S_2 \in \mathcal{P}^-(I)$.

Any $\cap$-concave set function, F, defines a monotone linkage, d_F, by

$$d_F(i,S) := \max_{S \subseteq T \subseteq I-i} F(T) \tag{5.3}$$

for any $S \in \mathcal{P}^-(I)$ and $i \in I - S$.

These functions are interconnected so that for any $\cap$-concave $F : \mathcal{P}^-(I) \rightarrow R$, $M_{d_F} = F$. The other way leads to a weaker property: for any monotone linkage d, $d_{M_d} \leq d$.

Given a monotone linkage function, $d(i, S)$, a series $(i_1, ..., i_N)$ is referred to as a *d-series* if $d(i_{k+1}, S_k) = \min_{i \in I - S_k} d(i, S_k) = M_d(S_k)$ for any starting set $S_k = \{i_1, ..., i_k\}$, $k = 1, ..., N - 1$. This definition describes the seriation algorithm as a greedy procedure for constructing a d-series starting with $i_1 \in I$: having defined S_k, take any i minimizing $d(i, S_k)$ over all $i \in I - S_k$ as i_{k+1}, $k = 1, ..., N-1$. A subset $S \in \mathcal{P}^-(I)$ will be referred to as a *d-cluster* if there exists a d-series, $s = (i_1, ..., i_N)$, such that S is a maximizer of $M_d(S)$ over all starting sets S_k of s. Greedily found, d-clusters play an important part in maximizing the associated $\cap$-concave set function, $F = M_d$. If, for a d-series $s = (i_1, i_2, ..., i_N)$, a subset $S \subset I$ contains i_1, and i_{k+1} is the first element in s not contained in S (for some $k = 1, ...N - 1$), then

$$F(S_k) = d(i_{k+1}, S_k) \geq d(i_{k+1}, S) \geq F(S)$$

where $S_k = \{i_1, ..., i_k\}$. In particular, if S is an inclusion-minimal maximizer of F (with regard to $\mathcal{P}^-(I)$), then $S = S_k$, that is, S is a d-cluster (Kempner, Mirkin, Muchnik, 1997).

All the minimal maximizers of a $\cap$-concave set function $F = M_d$ on $\mathcal{P}^-(I)$) for a monotone linkage d can be found by using the following three-step extended greedy procedure:

Extended Greedy Procedure

(A) For each $i \in I$, define d-series p_i greedily starting from i as its first element.

(B) For each d-series $p_i = (i_1 := i, i_2, ..., i_N)$, take T_i equal to its smallest starting set with $F(T_i) = \max_k d(i_{k+1}, S_k\})$.

(C) Among the non-coinciding minimal d-clusters T_i, $i \in I$, choose those maximizing F.

Moreover, every non-minimal maximizer of F is a union of the minimal maximizers. The reverse statement, in general, is not true: some unions of minimal maximizers can be non-maximizers. Also, the minimal clusters, though nonoverlapping, may cover only a part of I.

Example. Let us apply the extended greedy procedure to the holistic linkage function, $hl(i, S) = \sum_{k \in K} \min_{j \in S} |y_{ik} - y_{jk}|$, and table 1 considered as matrix X. Two of the hl-series produced are 1(1)2(2)3(3)4(1)5(0)6(0)7 and 7(1)6(1)4(2)5(2)3(0)2(0)1 (as started from rows 1 and 7, respectively). The values $hl(i_{k+1}, S_k)$ are put in the parentheses. We can see that the hl-cluster $S = \{1, 2, 3\}$ is the only maximizer of the minimum split function $M_{hl}(S)$; no maximizer starts with 7 (neither with 4 nor 5 nor 6). □

The problem of maximizing $\cap$-concave set functions is exponentially hard when they are oracle-defined: every set indicator function $F_A(S)$ $(A \subset I)$ which is equal to 0 when $S \neq A$ and 1 when $S = A$ is obviously $\cap$-concave (a note by V. Levit). However, $\cap$-concave set functions can be maximized greedy-wisely when they are defined in terms of monotone linkage. Thus, the monotone linkage format may well serve as an easy-to-interpret and easy-to-maximize input for dealing with $\cap$-concave set functions.

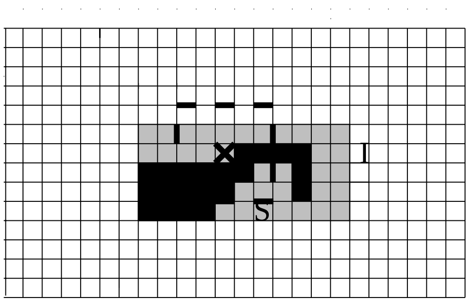

Figure 4: An illustration to the definition of the monotone-linkage function $l(i, S)$: I is the grey rectangle; S is the subcet of black cells; an $i \in I - S$ is the crossed cell being the center of a 5×5 window shown by the dashed borderline.

5.2.3 Modeling Skeletons in Digital Image Processing

In image analysis, there is a problem of skeletonization (or thinning) of planar patterns, that is, extracting a stick-like representation of a pattern to serve as a pattern descriptor for further processing (for a review, see Arcelli and Sanniti di Baja (1996)).

The linkage-based concave functions can suggest the set of inclusion-minimal maximizers as a skeleton model. This can be illustrated with the spatial data patterns in Figures 4 to 6.

On Fig. 4, the set of cells in the grey rectangle is I while the black cells constitute S. A cell $i \in I - S$ is in the center of a 5×5 window (shown by the dashed border line; its center cell, i, is shown by the cross); its linkage to S, $l(i, S)$ is defined as the number of grey cells in the window, which is obviously decreasing when S increases. It should be added to the definition that $l(i, S)$ is defined to be equal to 25 when the window does not overlap S; that is, when no black cell is present in the window around i.

In the example shown in Fig. 4, $l(i, S) = 11$. This can be decreased by moving i. The minimum value, $l(i, S) = 6$, is reached when the crossed cell is moved to the left border (within its row).

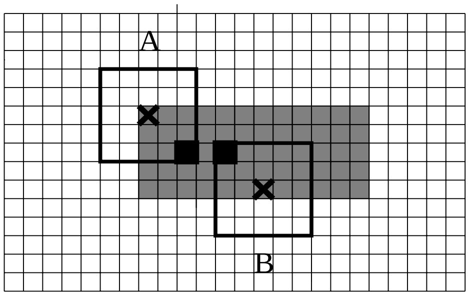

Figure 5: Two single-cell sets, A and B, illustrating different minimum values of $l(i, S)$ (over $i \in I - S$).

Obviously, set S must be a single cell to get the minimum of $l(i, S)$ (over all $i \in I - S$) maximally increased, as presented in Fig. 5.

The windows A and B represent the minimum values of $l(i, S)$ for each of the two subsets. The minimum value is 8 (for A) and 14 (for B), which makes B by far better than A with regard to the aim of maximizing $M_l(S)$. Obviously, $S = B$ is a minimal maximizer of $M_l(S)$. The set of minimal maximizers, for the given I, is the black skeleton strip as presented in Fig. 6.

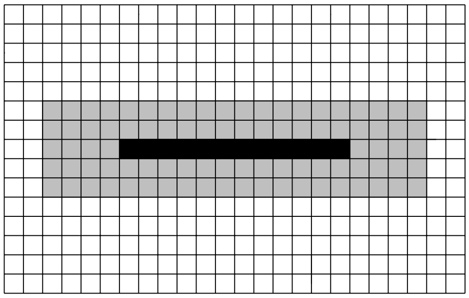

Figure 6: The strip of minimal maximizers of $M_l(S)$ being a skeleton of the grey I.

We leave it to the reader to analyze the changes of the set of minimal maximizers with respect to changes of the window size.

5.2.4 Linkage-based Convex Criteria

A $\cup$-concave set function $F(S)$ can be introduced via a monotone-increasing linkage function, that is, such a $p(i,S)$ with $i \in S$ that $p(i,S) \leq p(i, S \cup T)$ for any $T \subseteq I$. Let us define $D_p(S) = \min_{i \in S} p(i,S)$, the *diameter* of S. Obviously, $D_p(S) = M_{dp}(I-S)$ where $dp(i,S)$, $i \in I-S$, is a monotone-decreasing linkage function defined as $dp(i,S) = p(i, I-S)$. This implies that the diameter functions are those (and only those) satisfying the condition of $\cup$-concavity,

$$F(S_1 \cup S_2) \geq \min(F(S_1), F(S_2)) \tag{5.4}$$

Inequality (5.4) shows that the structure of maximizers, S^*, of a $\cup$-concave function is dual to the structure of maximizers, $I - S^*$, of the corresponding $\cap$-concave function. In particular, the set of maximizers of a $\cup$-concave function is closed with regard to union of subsets, and there exists only one inclusion-maximum maximizer for every $\cup$-concave function. This implies that a $\cap$-concave function may have several inclusion-minimal maximizers if and only if I itself is the only maximizer of the corresponding $\cup$-concave function.

Finding the maximum maximizer of a $\cup$-concave function D_p can be done greedily with a version of the seriation algorithm involving $p(i,S)$. Let us consider a p-series, $s = (i_1, ..., i_N)$, where every i_k is a minimizer of $p(i, I - S_{k-1})$ with regard to $i \in I - S_{k-1}$ ($k = 1, ..., N$; S_0 is defined as empty set). That $I - S_{k-1}$ is the maximum maximizer of $D_p(S)$ which gives maximum of $p(i_k, I - S_{k-1})$ ($k = 1, ..., N$). In this version, computation starts with I and goes on by one-by-one extracting entities from the set.

Two more kinds of set functions can be introduced dually, by switching between the operations of minimum and maximum (or just substituting the linkage functions d and p by $MM-d$ and $MM-p$ where MM is a constant). This way we obtain classes of what can be called $\cap$- and $\cup$-convex functions defined by conditions

$$F(S_1 \cap S_2) \leq \max(F(S_1), F(S_2)),\ F(S_1 \cup S_2) \leq \max(F(S_1), F(S_2)),$$

respectively. These functions are to be minimized. All the theory above remains applicable (up to obvious changes). An example of application of the convex functions for feature selection in regression problems has been provided by Muchnik and Kamensky (1993).

The monotone linkage functions were introduced, in the framework of clustering, by Mullat (1976) who considered $\cup$-convex functions $G(S) := \max_{i \in S} d(i, S)$ as greedily minimizable and called them "monotone systems". Constrained optimization clustering problems with this kind of functions were considered in Muchnik and Schwarzer (1989, 1990). This theory still needs to be polished. We will limit ourselves with an example based on the table 1 considered as the adjacency matrix for the graph in Fug. 2.

Example. Let us consider two monotone-increasing linkage functions,

$$p(i, S) = \sum_{j \in S} x_{ij} (\sum_{k \in S-j} x_{jk})^2$$

and

$$\pi(i, S) = \sum_{j \in S} x_{ij},$$

and define a constrained version of the diameter function,

$$D_{p\pi 3}(S) = \min_{i \in S \& \pi(i,S) \leq 3} p(i, S).$$

This function still satisfies the condition of $\cup$-concavity (5.4) and, moreover, can be maximized with the seriation algorithm above, starting with $S = I$ and one by one removing entities.

First step: Put $S = I$ and find set $\Pi(S) = \{i : i \in S \& \pi(i, S) \leq 3\}$ which is, obviously, $\Pi(I) = \{1, 3, 4\}$. Find $p(1, S) = 4+9+9 = 22$, $p(3, S) = 4+9+16 = 29$, and $p(4, S) = 16 + 16 + 9 = 41$. Thus, $D_{p\pi 3}(I) = 22$, and entity 1 is the first to be extracted so that, at the next step, $S := I - \{1\}$.

Second step: Determine $\Pi(S) = \{2, 3, 4\}$. Find $p(2, S) = 4 + 4 + 16 = 24$, $p(3, S) = 4 + 16 = 20$, and $p(4, S) = 16 + 16 + 9 = 41$. Thus, $D_{p\pi 3}(I - \{1\}) = 20$ and S becomes $S := I - \{1, 3\}$.

Third step: $\Pi(S) = \{2, 4\}$. Find $p(2, S) = 1+16 = 17$ and $p(4, S) = 16+9+9 = 34$. Thus, $D_{p\pi 3}(I - \{1, 2\}) = 17$ and S becomes $S := I - \{1, 2, 3\}$.

Fourth step: $\Pi(S) = \{4, 5\}$. We have $p(4, S) = p(5, S) = 9 + 9 + 9 = 27$, which makes $D_{p\pi 3}(I - \{1, 2, 3\}) = 27$ and S can be reduced by extracting either 4 or 5. Let, for instance, $S := I - \{1, 2, 3, 4\}$.

Fifth step: $\Pi(S) = \{5, 6, 7\} = S$. We have $p(5, S) = 4 + 4 = 8$ and $p(6, S) = p(7, S) = 4 + 4 + 4 = 12$. Thus, $D_{p\pi 3}(I - \{1, 2, 3, 4\}) = 8$ and next $S := \{6, 7\}$ which further reduces $D_{p\pi 3}(S)$.

Thus maximum $D_{p\pi 3}(S)$ is $D_{p\pi 3}(I - \{1, 2, 3\}) = 27$; the optimal S is the four-element set $S^* = \{4, 5, 6, 7\}$.

Curiously, this result is rather stable with regard to data changes. For instance, all the loops (diagonal ones) can be removed or added, with no change in the optimum cluster. □

The constructions described only involve ordering information in both, the domain and range of set/linkage functions and, also, they rely on the fact that every subset is uniquely decomposable into its elements. Therefore, they can be extended to distributive lattice structures with the set of irreducible elements as I (see, for instance, Libkin, Muchnik and Schwarzer (1989)).

5.3 Moving Center and Approximation Clusters

5.3.1 Criteria for Moving Center Methods

Let us say that a centroid concept, $c(S)$, *corresponds* to a dissimilarity measure, d, if $c(S)$ minimizes $\sum_{i\in S} d(i,c)$. For example, the gravity center (average point) corresponds to the squared Euclidean distance $d^2(y_i,c)$ since the minimum of $\sum_{i\in S} d^2(y_i,c)$ is reached when $c=\sum_{i\in S} y_{ik}/|S|$. Analogously, the median vector corresponds to the city-block distance.

For a subset $S\subset I$ and a centroid vector c, let us define

$$D(c,S)=\sum_{i\in S} d(i,c)+\sum_{i\in I-S} d(i,a) \tag{5.5}$$

to be minimized by both kinds of the variables (one related to c, the other to S). Here a is a reference point.

The alternating minimization of (5.5) consists of two steps reiterated: (1) given S, determine its centroid, c, by minimizing $\sum_{i\in S} d(i,c)$; (2) given c, determine $S=S(c)$ by minimizing $D(c,S)$ over all S. It appears, when the centroid concept corresponds to the dissimilarity measure, the moving center method is equivalent to the alternating minimization algorithm. In the case when the radius, r, is constant, the reference point a in (5.5) is set as a particular distinct point ∞ added to I with all the distances $d(i,\infty)$ equal to the radius r.

This guarantees convergence of the method to a locally optimal solution in a finite number of steps.

5.3.2 Principal Cluster

Criterion (5.5) can be motivated by the following approximation clustering model. Let $Y=(y_{ik})$, $i\in I$, $k\in K$, be an entity-to-variable data matrix. A cluster can be represented with its standard point $c=(c_k)$, $k\in K$, and a Boolean indicator function $s=(s_i)$, $i\in I$ (both of them may be unknown).

Let us define a bilinear model connecting the data and cluster with each other:

$$y_{ik} = c_k s_i + e_{ik},\ i \in I,\ k \in K, \tag{5.6}$$

where e_{ik} are residuals whose values show how well or ill the cluster structure fits into the data. The equations (5.6) when e_{ik} are small, mean that the rows of Y are of two different types: a row i resembles c when $s_i = 1$, and it has all its entries small when $s_i = 0$.

Consider the problem of fitting the model (5.6) with the least-squares criterion:

$$L^2(c,s) = \sum_{i \in I} \sum_{k \in K} (y_{ik} - c_k s_i)^2 \tag{5.7}$$

to be minimized with regard to binary s_i and/or arbitrary c_k.

A minimizing cluster structure is referred to as a *principal cluster* because of the analogy between this type of clustering and the principal component analysis: a solution to the problem (5.7) with no Booleanity restriction applied gives the principal component score vector s and factor loadings c corresponding to the maximum singular value of matrix Y (see, for data analysis terminology, Jain and Dubes (1988), Lebart, Morineau, and Piron (1995), Mirkin (1996)). It can be easily seen that criterion (5.7) is equivalent to criterion (5.5) with d being the Euclidean distance squared, $S = \{i : s_i = 1\}$, and $a = 0$, so that the moving center method entirely fits into the principal cluster analysis model (when it is modified by adding a given constant vector a to the right part). Obviously, changing $L^2(c,s)$ in (5.7) for other Minkowski criteria leads to (5.5) with corresponding Minkowski distances.

On the other hand, by presenting (5.7) in matrix form, $L^2 = Tr[(Y - sc^T)^T(Y - sc^T)]$, and putting there the optimal $c = Y^T s/s^T s$ (for s fixed), we have

$$L^2 = Tr(Y^T Y) - sYY^T s/s^T s$$

leading to decomposition of the square scatter of the data $(Y,Y) = Tr(Y^T Y) = \sum_{i,k} y_{ik}^2$ into the "explained" term, $sYY^T s/s^T s$, and the "unexplained" one, $L^2 = Tr(E^T E) = (E,E)$, where $E = (e_{ik})$:

$$(Y,Y) = sYY^T s/s^T s + (E,E) \tag{5.8}$$

Matrix $A = YY^T$ is an $N \times N$ entity-to-entity similarity matrix having its entries equal to the row-to-row scalar products $a_{ij} = (y_i, y_j)$. Let us denote the average similarity within a subset $S \subseteq I$ as $a(S) = \sum_{i,j \in S} a_{ij}/|S||S|$.

Then (5.8) implies that the principal cluster is a Boolean maximizer of the set function

$$g(S) = sYY^T s/s^T s = \frac{1}{|S|}\sum_{i,j\in S} a_{ij} = |S|a(S) \qquad (5.9)$$

which extends the concept of subgraph density function onto arbitrary Gram matrices $A = YY^T$.

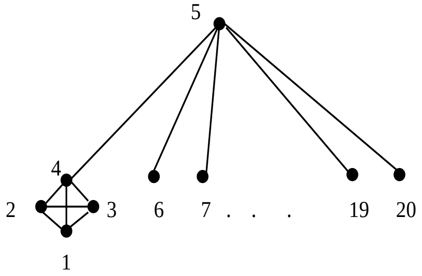

Figure 7: A graph with a four-element clique which is not the first choice in the corresponding eigenvector.

As is well known, maximizing criterion $sYY^T s/s^T s$ with no constrains on s yields the optimal s to be the eigenvector of A corresponding to its maximum eigenvalue (Janich (1994)). This may make one suggest that there must be a correspondence between the components of the globally optimal solution (the eigenvector) and the solution to the restricted problem when s is to be Boolean. However, even if such a correspondence exists, it is far from being straightforward. For example, there is no correspondence between the largest components of the eigen-vector and the non-zero components in the optimal Boolean s: the first eigen-vector for the 20-vertex graph in Fig. 7 has its maximum value corresponding to vertex 5 which, obviously, does not belong to the maximum density subgraph, the clique $\{1,2,3,4\}$.

Let us consider a local search algorithm for maximizing $g(S)$ starting with $S = \emptyset$ and adding entities from I one by one. This means that the algorithm exploits the neighborhood $N(S) := \{S + i : i \in I - S\}$ and is a seriation algorithm based on the increment $\delta g(i,S) := g(S+i) - g(S)$ which is

$$\delta g(i,S) = \frac{a_{ii} + 2|S|al(i,S) - g(S)}{|S|+1} \qquad (5.10)$$

where $al(i,S)$ is the average linkage function defined above as $al(i,S) := \sum_{i,j\in S} a_{ij}/|S|$.

A dynamic version of the seriation algorithm with this linkage function is as follows.

Local Search for g(S)

At every iteration, the values $Al(i,S) = a_{ii}+2|S|al(i,S)$ $(i \in I-S)$ are calculated and their maximum $Al(i^*,S)$ is found. If $Al(i^*,S) > g(S)$ then i^* is added to S; if not, the process stops, S is the resulting cluster.

To start a new iteration, all the values are recalculated:
$al(i,S) \Leftarrow (|S|al(i,S) + a_{ii^*})/(|S|+1)$
$g(S) \Leftarrow (|S|g(S) + Al(i^*,S))/(|S|+1)$
$|S| \Leftarrow |S|+1$.

Since criteria (5.9) and (5.5) (with $a = 0$) are equivalent, the results of these two seemingly different techniques, the seriation and moving centers procedures, usually will be similar.

Example. The algorithm applied to the data in table 2 produces a two-entity principal cluster, $S = \{1,2\}$, whose contribution to the data scatter is 32.7%. Reiterating the process to the set of yet unclustered entities we obtain a partition whose classes are in table 9. The clusters explain some 85% of the data scatter.

Table 9: The sequential principal clusters for 7 entities by the column-conditional matrix 2.

Cluster	Entities	Contribution, %
1	1, 2	32.7
2	3	13.6
3	4, 6, 7	26.0
4	5	12.4

□

5.3.3 Additive Cluster

Let $A = (a_{ij})$, $i,j \in I$, be a given similarity or association matrix and λs=$(\lambda s_i s_j)$ a weighted set indicator matrix which means that $s = (s_i)$ is the indicator of a subset $S \subseteq I$ along with its intensity weight λ. The following model is applicable when A can be considered as a noisy information on $\lambda\mathbf{s}$:

$$a_{ij} = \lambda s_i s_j + e_{ij} \tag{5.11}$$

where e_{ij} are the residuals to be minimized. Usually, matrix A must be centered (thus having zero as its grand mean) to make the model look fair.

The least-squares criterion for fitting the model,

$$L^2(\lambda, s) = \sum_{i,j \in I} (a_{ij} - \lambda s_i s_j)^2, \tag{5.12}$$

is to be minimized with regard to unknown Boolean $s = (s_i)$ and/or real λ (in some problems, λ may be predefined). When λ is not subject to change, the criterion can be presented as

$$L^2(\lambda, s) = \sum_{i,j \in I} a_{ij}^2 - 2\lambda \sum_{i,j \in I} (a_{ij} - \lambda/2) s_i s_j$$

which implies that, for $\lambda > 0$ (which is assumed for the sake of simplicity), the problem in (5.12) is equivalent to maximizing

$$L(\lambda/2, s) = \sum_{i,j \in S} (a_{ij} - \lambda/2) = \sum_{i,j \in S} a_{ij} - \lambda/2|S|^2 \tag{5.13}$$

which is just the summary threshold linkage criterion, $L(\pi, S) = \sum_{i \in S} l_\pi(i, S)$ where $\pi = \lambda/2$. This implies that the seriation techniques based on $l_\pi(i, S)$ can be applied for locally maximizing (5.13).

In general, the task of optimizing criterion (5.13) is NP-complete.

Let us now turn to the case when λ is not pre-defined and may be adjusted based on the least-squares criterion. There are two optimizing options available here.

The first option is based on the representation of the criterion as a function of two variables, S and λ, to allow using the alternating optimization technique.

Alternating Optimization for Additive Clustering
Each iteration includes: first, finding a (locally) optimal S for $L(\pi, S)$ with $\pi = \lambda/2$; second, determining the optimal $\lambda = \lambda(S)$, for fixed S, by the formula below. The process ends when no change of the cluster occurs.

The other option is based on another form of the criterion. For any given S, the optimal λ can be determined (by making derivative of $L^2(\lambda, s)$ by λ equal to zero) as the average of the similarities within S:

$$\lambda(S) = a(S) = \sum_{i,j \in I} a_{ij} s_i s_j / \sum_{i,j \in I} s_i s_j$$

The value of L^2 in (5.12) with the $\lambda = \lambda(S)$ substituted becomes:

$$L^2(\lambda, s) = \sum_{i,j} a_{ij}^2 - (\sum_{i,j} a_{ij} s_i s_j)^2 / \sum_{i,j} s_i s_j \tag{5.14}$$

Since the first item in the right part is constant (just the square scatter of the similarity coefficients), minimizing L^2 is equivalent to maximizing the second item which is the average linkage criterion squared, $g^2(S)$ where $g(S)$ is defined in (5.9). Thus, the other option is just maximizing this criterion with the local search techniques described.

Some more comments:

(1) the criterion is the additive contribution of the cluster to the square scatter of the similarity data, which can be employed to judge how important the cluster is (in its relation to the data);

(2) since the function $g(S)$ here is squared, the optimal solution may correspond to the situation when $g(S)$ is negative, as well as $L(\pi, S)$ and $a(S)$. Since the similarity matrix A normally is centered, that means that such a subset consists of the most disassociated entities and should be called anti-cluster. However, using local search algorithms allows us to have that sign of $a(S)$ we wish, either positive or negative: just the initial extremal similarity has to be selected from only positive or only negative values;

(3) in a local search procedure, change of the squared criterion when an entity is added/removed may behave slightly differently than that of the original $g(S)$ (an account of this is done in Mirkin (1990));

(4) when $A = YY^T$ where Y is a column-conditional matrix, the additive cluster criterion is just the principal cluster criterion squared, which implies that the optimizing clusters must be the same, in this case.

5.3.4 Seriation with Returns

Considered as a clustering algorithm, the seriation procedure has a drawback: every particular entity, once being caught in the sequence, can never be relocated, even when it has low similarities to the later added elements. After the optimization criteria have been introduced, such a drawback can be easily overcome. To allow exclusion of the elements in any step of the seriation process, the algorithm is modified by extending its neighborhood system.

Let, for any $S \subset I$, its neighborhood $N(S)$ consist of all the subsets differing from S by an entity $i \in I$ being added to or removed from S. The local search techniques can be formulated for any criterion as based on this modification. In particular, criterion $g(S)$ has its increment in the new $N(S)$ equal to

$$\delta g(i,S) = \frac{a_{ii} + 2z_i|S|al(i,S) - z_i a(S)}{|S| + z_i} \tag{5.15}$$

where $z_i = 1$ if i has been added to S or $z_i = -1$ if i has been removed from S. Thus, the only difference between this formula and that in (5.10) is change of the sign in some terms. This allows for a modified algorithm.

Local Search with Return for g(S)

At every iteration, values $Al(i,S) = s_{ii} + 2z_i|S|al(i,S)$ $(i \in I)$ are calculated and their maximum $Al(i^*,S)$ is found. If $Al(i^*,S) > z_{i^*}g(S)$, then i^* is added to or removed from S by changing the sign of z_{i^*}; if not, the process stops, S is the resulting cluster.

To start the next iteration, all the values are updated:

$al(i,S) \Leftarrow (|S|al(i,S) + z_{i^*}s_{ii^*})/(|S| + z_{i^*})$

$g(S) \Leftarrow (|S|g(S) + z_{i^*}Al(i^*,S))/(|S| + z_{i^*})$

$\nu(S) \Leftarrow |S| + z_{i^*}$.

The cluster found with the modified local search algorithm is a strict cluster since the stopping criterion involves the numerator of (5.15) and implies inequality $z_i(al(i,S) - g(S)/2|S|) \leq 0$ for any $i \in I$.

6 Partitioning

6.1 Partitioning Column-Conditional Data

6.1.1 Partitioning Concepts

There are several different approaches to partitioning:

A *Cohesive Clustering*: (a) within cluster similarities must be maximal; (b) within cluster dissimilarities must be minimal; (c) within cluster dispersions must be minimal.

B *Extreme Type Typology*: cluster prototypes (centroids) must be as far from grand mean as possible.

C *Correlate (Consensus) Partition*: correlation (consensus) between the clustering partition and given variables/categories must be maximal.

D *Approximate Structure*: differences between the data matrix and cluster structure matrix must be minimal.

There can be suggested an infinite number of criteria within each of these loose requirements. Among them, there exists a list of criteria that are equivalent to each other. This fact and numerous experimental results are in favor of the criteria listed in the following four subsections.

6.1.2 Cohesive Clustering Criteria

Let $S = \{S_1, ..., S_m\}$ be a partition of I to be found with regard to the data given. A criterion of within cluster similarity to maximize:

$$g(S) = \sum_{t=1}^{m} \sum_{i,j \in S_t} a_{ij}/|S_t| = \sum_t g(S_t) \tag{6.1}$$

where $A = (a_{ij})$ is a similarity matrix.

A criterion of within cluster dissimilarity to minimize:

$$D(S) = \sum_{t=1}^{m} \sum_{i,j \in S_t} d_{ij}/|S_t| \tag{6.2}$$

where $D = (d_{ij})$ is a dissimilarity matrix.

Two criteria of within cluster dispersion to minimize:

$$D(c, S) = \sum_{t=1} \sum_{i \in S_t} d(c_t, y_i) \tag{6.3}$$

where $d(c_t, y_i)$ is a dissimilarity measure between the row-point y_i, $i \in I$, and cluster centroid c_t; and

$$\sigma(S) = \sum_{t=1}^{m} p_t \sigma_t^2 \tag{6.4}$$

where $p_t = |S_t|/|I|$ is proportion of the entities in S_t and $\sigma_t = \sum_k \sum_{i \in S_t} (y_{ik} - c_{tk})^2/|S_t|$ is the total variance in S_t with regard to within cluster means, $c_{tk} = \sum_{i \in S_t} y_{ik}/|S_t|$.

The versions of criteria (6.1), (6.2) and (6.4) with no cluster coefficients p_t, $|S_t|$ have been also considered.

6.1.3 Extreme Type Typology Criterion

A criterion to maximize is

$$T(c,S) = \sum_{v \in V} \sum_{t=1}^{m} c_{tv}^2 |S_t| \tag{6.5}$$

where c_{tv} is the average of the category/variable v in S_t.

6.1.4 Correlate/Consensus Partition

A criterion to maximize is

$$C(S) = \sum_{k \in K} \mu(S,k) \tag{6.6}$$

where $\mu(S,k)$ is a correlation or contingency coefficient (index of consensus) between S and a variable $k \in K$. Important examples of such coefficients:

(i) Correlation ratio (squared) when k is quantitative,

$$\eta^2(S,k) = \frac{\sigma_k^2 - \sum_{t=1}^{m} p_t \sigma_{tk}^2}{\sigma_k^2} \tag{6.7}$$

where p_t is the proportion of entities in S_t, and σ_k^2 or σ_{tk}^2 is the variance of variable k in all the set I or within cluster S_t, respectively. Correlation ratio $\eta^2(S,k)$ is between 0 and 1; $\eta^2(S,k) = 1$ if and only if variable k is constant in each cluster S_t.

(ii) Pearson X^2 (chi-square) when k is qualitative,

$$X^2(S,k) = \sum_{v \in k} \sum_{t=1}^{m} \frac{(p_{vt} - p_v p_t)^2}{p_v p_t} = \sum_{v \in k} \sum_{t=1}^{m} \frac{p_{vt}^2}{p_v p_t} - 1 \tag{6.8}$$

where p_v, p_t, or p_{vt} are frequencies of observing of a category v (in variable k), cluster S_t, or both. Actually, the original Pearson coefficient was introduced as a measure of deviation of observed bivariate distribution, p_{vt}, from the hypothetical statistical independence: it is very well known in statistics that when the deviation is due to sampling only, distribution of NX^2 is asymptotically chi-square. However, in clustering this index may have different meanings (Mirkin (1996)).

(iii) Reduction of the proportional prediction error when k is qualitative,

$$\Delta(S/k) = \sum_{v \in k} \sum_{t=1}^{m} \frac{(p_{vt} - p_v p_t)^2}{p_t} = \sum_{v \in k} \sum_{t=1}^{m} p_{vt}^2 / p_t - \sum_{v \in k} p_v^2 \tag{6.9}$$

The proportional prediction error is defined as probability of error in the so-called proportional prediction rule applied to randomly coming entities when any v is predicted with frequency p_v. The average error of proportional prediction is equal to $\sum_v p_v(1-p_v) = 1 - \sum_v p_v^2$ which is also called Gini coefficient. $\Delta(S/k)$ is reduction of the error when the proportional prediction of v is made under condition that t is known.

The coefficient $C(S)$ with $\mu(S,k) = \Delta(S/k)$ is proven in Mirkin (1996) to be equivalent to yet another criterion to maximize which is frequently used in conceptual clustering, the so-called Category Utility Function applied only when all the variables are nominal (see, for instance, Fisher (1987)).

6.1.5 Approximation Criteria

Let the data be represented as a data matrix $Y = (y_{iv}), i \in I, v \in V$, where rows $y_i = (y_{iv}), v \in V$, correspond to the entities $i \in I$, columns, v, to quantitative variables or qualitative categories, and the entries y_{iv} are quantitative values associated with corresponding variables/categories $v \in V$. A category v is represented in the original data table by a binary column vector with unities assigned to entities satisfying v and zeros to the others. The sizes of these sets will be denoted, as usual: $|I| = N$ and $|V| = n$.

To present a partition $S = \{S_1, ..., S_m\}$ as a matrix of the same size, let us assign centroids, $c_t = (c_{tv})$, to clusters S_t presented by corresponding binary indicators, $s_t = (s_{it})$, where $s_{it} = 1$ if $i \in S_t$ and $= 0$ if $i \notin S_t$. Then, the matrix $\sum_t s_t c_t^T$ represents the cluster structure so that comparison of the structure and the data can be done via equations

$$y_{iv} = \sum_{t=1}^{m} c_{tv} s_{it} + e_{iv} \tag{6.10}$$

where e_{iv} are residuals to be minimized by the cluster structure using, for instance, the least-squares criterion:

$$L_2(S,c) = \sum_{i \in I} \sum_{v \in V} (y_{iv} - \sum_{t=1}^{m} c_{tv} s_{it})^2 \tag{6.11}$$

Since S is a partition, a simpler formula holds for the criterion:

$$D_2(S,c) = \sum_{t=1}^{m} \sum_{v \in V} \sum_{i \in S_t} (y_{iv} - c_{tv})^2 \tag{6.12}$$

6.1.6 Properties of the Criteria

Let us assume that a standardizing transformation (2.1) has been applied to each variable/category $v \in V$ with a_v the grand mean and b_v the standard deviation if v is a quantitative variable. When v is a binary category, a_v is still the grand mean equal to the frequency of v, p_v. For b_v, one of the following two options is suggested: $b_v = 1$ (first option) or $b_v = \sqrt{p_v}$ (second option). Then the following statement holds.

The following criteria are equivalent to each other:

(6.1) with $a_{ij} = (y_i, y_j)$, that is, $A = YY^T$,

(6.2) with d_{ij} being Euclidean distance squared,

(6.3) with $d(y_i, c_t)$ being Euclidean distance squared,

(6.4),

(6.5),

(6.6) where $\mu(S,k) = \eta(S,k)$ if k is quantitative and $\mu(S,k) = \Delta(k/S)$ if k is qualitative and the first standardizing option has been applied or $\mu(S,k) = \chi^2(S,k)$ if k is qualitative and the second standardizing option has been applied,

(6.11), and

(6.12).

The proof can be found in Mirkin (1996); it is based on the following decomposition of the data scatter in model (6.10) when c_{tv} are least-squares optimal (thus being the within cluster means):

$$\sum_{i \in I} \sum_{v \in V} y_{iv}^2 = \sum_{t=1}^{m} \sum_{v \in V} c_{tv}^2 |S_t| + \sum_{i \in I} \sum_{v \in V} e_{iv}^2, \qquad (6.13)$$

Those of the criteria to be maximized correspond to the "partition-explained" part of the data scatter and those to be minimized correspond to the "unexplained" residuals.

Some properties of the criteria:

(1) The larger m, the better value. This implies that either m or a criterion value (as the proportion of the "explained" data scatter) should be predefined as a stopping criterion.

(2) When c_t are given, the optimal clusters S_t satisfy the so-called *minimal distance rule*: for every $i \in S_t$, $d(y_i, c_t) \le d(y_i, c_q)$ for all $q \neq t$. This means that the optimal clusters are within nonoverlapping balls (spheres) that are convex bodies. This drastically reduces potential number of candidate partitions in enumeration algorithms. In the case of $m = 2$, the optimal clusters must be linearly separated. By shifting the separating hyperplane

toward one of the clusters until it touches an entity of I, we get a number of the entity points belonging to the shifted hyperplane. Since the total number of the points defining the normal vector is $|V|$, the total number of the separating hyperplanes is not larger than $\left(\begin{array}{c} 1 \\ N \end{array} \right) + \left(\begin{array}{c} 2 \\ N \end{array} \right) + ... + \left(\begin{array}{c} |V| \\ N \end{array} \right) \leq N^{|}V|$. This guarantees a "polynomial"-time solution to the problem by just enumeration of the separating hyperplanes. Regretfully, there is nothing known on the problem beyond that. Probably an m cluster optimal partition can be found by enumerating not more than $N^{|V|m/2}$ separating hyperplanes.

(3) The criterion (6.1) is an extension of the maximum density subgraph problem; this time the total of within cluster densities must be maximized.

(4) Different expressions differently fit into different neighborhoods for local search. For instance, formula (6.12) fits into alternating minimization strategy (given c, adjust S; given S, adjust c). Formula (6.6) is preferable in conceptual clustering when partitioning is done by consecutive dividing I by the variables.

6.1.7 Local Search Algorithms

Among many clustering heuristics suggested, those seem to have better chances for survival that: (a) are local search algorithms for convenient criteria, and (b) can be interpreted as models of a human classification making process. We present here several partition neighborhood systems applicable to any criteria.

Agglomerative Clustering. This procedure models establishing a biological taxonomy via similarity of species. The neighborhood of a partition $S = \{S_1, ..., S_m\}$ here is $N(S) = \{S^{tu} : t, u = 1, ..., m; t \neq u\}$ where S^{tu} is obtained from S by merging its classes S_t and S_u. For criterion (6.11), the local search algorithm with this neighborhood system can be formulated as follows (starting with the matrix of Euclidean distances squared):

Agglomerative Clustering

Step 1. Find the minimal value $d_{i^*j^*}$ in the dissimilarity matrix and merge $i^{''}$ and j^* .
Step 2. Reduce the distance matrix, substituting one new row (and column) $i^* \cup j^*$ instead of the rows and columns i^*, j^*, with its dissimilarities defined as

$$d_{i,i^*\cup j^*} = |S_t||S_u|/(|S_t| + |S_u|)d^2(c_t, c_u) \qquad (6.14)$$

where c_t and c_u are the cluster gravity centers. If the number of clusters is larger than 2, go to Step 1, else End.

The value $d_{i,i^*\cup j^*}$ is exactly the increment of criterion D_2 (6.12) when S is changed for S^{tu} (Ward (1963)). The algorithm is known as Ward clustering algorithm.

Lance and Williams (1967) suggested a family of agglomerative clustering algorithms by extending equation (6.14) to a linear-wise function of the former distances d_{i,i^*} and d_{i,j^*}. Among the most popular in the Lance-Williams family are the single linkage and complete linkage agglomerative clustering algorithms where $d_{i,i^*\cup j^*} = \min(d_{i,i^*}, d_{i,j^*})$ and $d_{i,i^*\cup j^*} = \max(d_{i,i^*}, d_{i,j^*})$, respectively. Ward algorithm also belongs to this family.

Alternating Square-Error Clustering. This algorithm can be considered a model for typology making. It corresponds also to the most popular technique in clustering, the so-called K-Means (moving centers, nuée dynamique) method for direct clustering.

Alternating Square-Error Clustering
Starting with a list of tentative centers c_t, the following two steps are iterated until the partition is stabilized:
Step 1. *Cluster membership.* Having c_t fixed, find clusters S_t minimizing $\sum_{t=1}^{n}\sum_{i\in S_t} d^2(y_i, c_t)$ with the minimal distance rule.
Step 2. *Standard points.* Having clusters S_t fixed, find the gravity centers, c_t, of S_t, $t = 1, ..., m$.

The method converges in a finite number of iterations since the number of minimal distance rule partitions is finite and the criterion decreases at each step. The other versions of the algorithm involve different dissimilarity measures and centroid concepts.

Exchange Algorithm. This algorithm can be considered as a procedure for one-by-one correcting a predefined partition S. Its neighborhood system can be defined as $N(S) = \{S(i,t) : i \in I, t = 1, ..., m\}$ where $S(i,t)$ is a partition obtained from S by putting entity i into class S_t.

```
Exchange Algorithm
Step 0. Take the initial $S$.
Step 1. Find a pair $(i^*, t^*)$ maximizing the criterion value in $N(S)$.
Step 2. If the criterion value is better than that for $S$, put $S \leftarrow S(i^*, t^*)$
and go to Step 1. Otherwise end.
```

To make the algorithm more time-efficient, an order on the entity set can be specified so that, at each iteration, only one i is considered at Step 1; the next iteration deals with the next entity, and so on (the first entity goes after the last one), until all the entities are tried a predefined number of times.

Another, quite popular, version of the exchange procedure is exploited in those applications in which the cluster cardinalities are not supposed to be varied. In this case, the neighborhood system is $N(S) := \{S(i,j) : i, j \in I\}$ where $S(i,j)$ is a partition obtained from S by switching the entities i and j between their classes. Again, a prespecified ordering of I can reduce computations so that at each iteration only one i is taken according to the ordering (next iteration, next entity). In the problems of dividing a graph in two even parts (bisection problems), this version is well known as Kernighan-Lin (1970) heuristics. In the clustering research, it was known somewhat earlier (see, for instance, a review by Dorofeyuk (1971) referring to a work of one of the authors published in 1968, in Russian).

Conceptual Clustering

The conceptual clustering algorithms construct decision trees divisively from top to bottom (with the root to represent the universe considered) using one variable at each step so that the following problems are addressed:

1. Which class (node of the tree) and by which variable to split?

2. When to stop splitting?

3. How to prune/aggregate the tree if it becomes too large?

4. Which class to assign to a terminal node?

Traditionally, conceptual clustering is considered as an independent set of machine learning techniques. However, in the framework of this presentation, at least some of conceptual clustering techniques can be considered as yet other local search procedures. To decide which class S to split and by which variable it is to be split, the goodness-of-split criterion (6.6) implied by the theory above can be utilized.

Example. The tree in Fig. 8 shows results of chi-square based criterion (6.8) applied to the task of splitting the set of seven entities by the data in table 1 considered as binary categories.

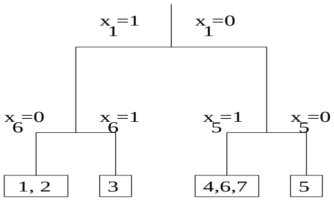

Figure 8: A conceptual tree for the binary data in table 1 found with the summary chi-square coefficient as the goodness-of-split criterion.

□

6.2 Criteria for Similarity Data

6.2.1 Uniform Partitioning

A similarity matrix $A = (a_{ij})$ given, this clustering model can be expressed with equation $A = \lambda SS^T + E$ where λ is a real and $S = (s_{it})$ is the indicator matrix of a partition. Minimizing the least-squares criterion, $(A-\lambda SS^T, A-\lambda SS^T)$, is equivalent to maximizing $a_w^2 n_w$ where a_w and n_w are respectively the average and number of similarities within all clusters. This problem will be referred to as the uniform partitioning model. No number of clusters m prespecified is needed in this model.

With λ predefined (for the sake of brevity, assume $\lambda > 0$), the uniform partitioning criterion is equivalent to

$$F(S,\lambda) = \sum_{t=1}^{m} \sum_{i,j \in S_t} (a_{ij} - \lambda/2) = \sum_{t=1}^{m} L(\lambda/2, S_t) \tag{6.15}$$

to be maximized, where $L(\lambda/2, S_t)$ is a single cluster criterion (5.13). Value $\lambda/2$ is a "soft" similarity threshold requiring that, in general, the larger similarities fall within the clusters, and the smaller similarities, between the clusters. In general, the larger λ, the smaller clusters, though it is not always so. There exists an index of the optimal partition closely following changes of λ: the larger λ, the smaller the error of proportional prediction (Gini index), $1-\sum_{t=1}^{m} p_t^2$, of the optimal partition (Kupershtoh, Mirkin, and Trofimov (1976)).

When the data matrix is not symmetric, $a_{ij} \neq a_{ji}$, it can be symmetrized with transformation $a_{ij} \leftarrow (a_{ij}+a_{ji})/2$ since criterion (6.15) is invariant with regard to this transformation. The transformation is applicable anytime when the approximating clustering structure is symmetric.

Example. The optimal uniform partitions of 10 digits for different thresholds $\lambda/2$ are presented in table 10. The Confusion data matrix has been preliminarily symmetrized; its diagonal entries have been excluded.

Table 10: Uniform partitions of 10 segmented digits (the Confusion data set).

Threshold	m	Partition	Residual Variance
-20	2	1-4-7, 2-3-5-6-8-9-0	0.754
0	4	1-4-7, 2, 3-5-9, 6-8-0	0.476
30	6	1-7, 2, 3-9, 4, 5-6, 8-10	0.439
50	6	1-7, 2, 3-9, 4, 5-6, 8-10	0.439
60	7	1-7, 2, 3-9, 4, 5, 6, 8-10	0.468
90	8	1-7, 2, 3-9, 4, 5, 6, 8, 10	0.593

□

The least-squares optimal λ is the within-partition average similarity, a_w.

This problem can also be addressed in the framework of alternating optimization: (1) reiteration of the steps of optimization of criterion (6.15), with λ fixed, and (2) calculation of the within-class-average λ, for the partition found.

Rationales for considering the uniform partitioning problem include the following (Mirkin (1996)):

(1) In an optimal partition, the average within class similarity is not larger than $\lambda/2$, and the average between class similarity is not smaller than $\lambda/2$. This gives an exact meaning of "clusterness" to the uniform partition classes.

(2) In a thorough experimental study, G. Milligan (1981) has demonstrated that the usual correlation coefficient between A and SS^T belongs to the best goodness-of-fit indices of clustering results. On the other hand, the coefficient characterizes quality of the matrix (bi)linear regression model, $A = \lambda SS^T + \mu U + E$ (where U is the matrix with all its entries equal to

unity). Therefore, the experimental results may be considered as those justifying use of the latter model as a clustering model; the uniform partitioning problem is just a shortened version of it. Curiously, this shortened version better fits into the cluster-wise meaning of the optimal partition than the original matrix regression model.

(3) Criterion (6.15) appears to be equivalent to that of the index-driven consensus problem in various settings. (A partition S is called an index-driven consensus partition if it maximizes $\sum_{k=1}^{n} \xi(S^k, S)$ where $S^1, ..., S^n$ are some given partitions on I and $\xi(S^k, S)$ is a between-partition correlation index.) In particular, it is true for the index being the number of noncoinciding edges in corresponding graphs (Hamming distance between adjacency matrices). This way the problem of approximation of a graph by a graph consisting of cliques (Zahn (1963)) fits within this one.

(4) In the context of the Lance-Williams agglomerative clustering, the uniform partitioning criterion appears to be the only one leading to the flexible Lance-Williams algorithms (with constant coefficients) as the optimization ones. We refer to an agglomerative clustering algorithm as an optimization one if its every agglomeration step, merging S_u and S_v into $S_u \cup S_v$, maximizes increment, $\delta_F(u, v) = F(S_u \cup S_v) - F(S_u) - F(S_v)$, of a set function F.

(5) Criterion (6.15) extends that of graph partitioning in 6.2.4 (when threshold is zero) providing also a compromise to the requirement of getting a balanced partition (the larger the threshold, the more uniform are the cluster sizes as measured by the Gini index).

6.2.2 Additive Partition Clustering

To fit into situations when real-world cluster patterns may show a great difference in cluster "diameters", a model with the clusters having distinct intensity weights can be considered (Shepard and Arabie (1979), Mirkin (1987)):

$$a_{ij} = \sum_{t=1}^{m} \lambda_t s_{it} s_{jt} + e_{ij} \qquad (6.16)$$

where e_{ij} are the residuals to be minimized.

In matrix terms, the model is $A = S\Lambda S^T + E$ where S is the $N \times m$ matrix of cluster indicator functions, Λ is the $m \times m$ diagonal matrix of λs, and $E = (e_{ij})$.

When the clusters are assumed mutually nonoverlapping (that is, the indicator functions s_t are mutually orthogonal) or/and when fitting of the model is made with the sequential fitting SEFIT strategy (see section 6.3), the data scatter decomposition holds as follows:

$$(A,A) = \sum_{t=1}^{m} [s_t A s_t^T / s_t^T s_t]^2 + (E,E) \tag{6.17}$$

where the least-squares optimal λ_ts have been put as the within cluster averages of the (residual) similarities.

It can be seen, from (6.17), that the least-squares fitting of the additive clustering model (under the nonoverlapping assumption) requires maximizing of the intermediate term in (6.17), which differs from (6.1) only in that the terms are squared here. No mathematical results on maximizing this criterion are known.

6.2.3 Structured Partitioning

Let $B = (b_{ij})$ be a matrix of association or influence coefficients and (S, ω) a structured partition in which ω is a digraph of "important" between-class associations. Such a structured partition can be represented by the Boolean matrix $\mathbf{S}_\omega = (s_{ij})$ where $s_{ij} = 1$ if and only if $(t,u) \in \omega$ for $i \in S_t$ and $j \in S_u$. Then, the linear model of the associations approximated by $\mathbf{S}_\omega$, is:

$$b_{ij} = \lambda s_{ij} + e_{ij} \tag{6.18}$$

This model suggests uniting in the same class those entities that identically interact with the others (being perhaps non associated among themselves).

The problem can be thought of as that of approximation of a large graph, B, by a smaller graph, (S,ω). This smaller graph is frequently called blockmodel in social psychology (Arabie, Boorman, and Levitt (1978)). The approximation model (6.18) is considered in Mirkin (1996).

When λ is positive, the least squares fitting problem for model (6.18) can be equivalently represented as the problem of maximizing

$$SU(\pi, S, \omega) = \sum_{(u,t)\in\omega} \sum_{i\in S_t} \sum_{j\in S_u} (b_{ij} - \pi) \tag{6.19}$$

by (S,ω) for $\pi = \lambda/2$.

When there is no constraints on ω and S is fixed, the optimal ω (for given S) can be easily identified depending on the summary proximity values

$$b(\pi, t, u) = \sum_{i\in S_t} \sum_{j\in S_u} (b_{ij} - \pi).$$

The structure ω maximizing (6.19) for given S is

$$\omega(S) = \{(t,u) : b(\pi,t,u) > 0\}.$$

This implies that, with no constraints on ω, maximizing criterion (6.19) is equivalent to maximizing criterion

$$AS(\pi, S) = \sum_{t,u=1}^{m} |b(\pi,t,u)|. \tag{6.20}$$

which doesn't depend on ω and, thus, can be locally optimized by local search algorithms (Kupershtoh and Trofimov, 1975).

Optimizing threshold λ when S is given can be done with the aggregate matrix $(b(0,t,u))$.

Example. Let us consider the Confusion data (between 10 segmented integer digits) from Table 6, p. 10, with the diagonal entries eliminated. The matrix B centered by subtracting its grand mean, 33.4556 (with no diagonal entries), is as follows:

−	−26.5	−26.5	−11.5	−29.5	−18.5	26.5	−33.5	−29.5	−29.5
−19.5	−	13.5	−29.5	2.5	13.5	−19.5	−4.5	−26.5	−15.5
−4.5	−4.5	−	−26.5	−15.5	−33.5	6.5	−4.5	118.5	−18.5
115.5	−11.5	−29.5	−	−29.5	−22.5	−3.5	−26.5	7.5	−33.5
−19.5	−7.5	9.5	−19.5	−	45.5	−26.5	−26.5	92.5	−19.5
−8.5	−19.5	−26.5	−22.5	63.5	−	−29.5	121.5	−22.5	9.5
235.5	−29.5	−12.5	−12.5	−26.5	−33.5	−	−33.5	−29.5	−26.5
−22.5	−5.5	−5.5	−15.5	−15.5	36.5	−22.5	−	33.5	138.5
−8.5	−4.5	77.5	12.5	48.5	−22.5	−12.5	48.5	−	9.5
−15.5	−29.5	−26.5	−22.5	−26.5	−15.5	−8.5	37.5	−12.5	−
1	2	3	4	5	6	7	8	9	0

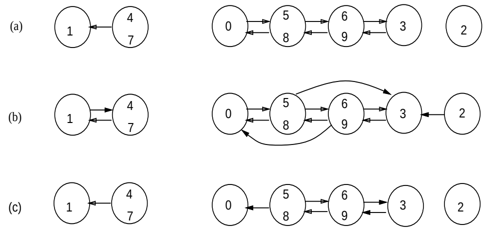

Figure 9: Structured partitions for the Confusion data.

An intuitively defined aggregate graph of major confusions between the digits is presented in Fig.9 (a): the non-singleton classes, 4-7, 5-8, and 6-9, unite unconnected entities. The structure comprises 18 entries in A, some of them being negative, such as, for instance, $a_{05} = -26.5$. This structure is not optimal, for $\pi = 0$. The optimal structure, $\omega(S)$ with $\pi = 0$, must include more associations as shown in Fig.9 (b). The average of all the 25 within structure entries is equal to 46.3, which makes $\pi = 23.15$ to cut out of the structure the weakest connections, such as from 2 to 3, with $a_{23} = 13.5 < \pi$. The final structured partition is in (c) ($\lambda = 68.4$) differing from (a) by just only one arrow deleted. □

In Mirkin (1996) an interpretation of a similar approximation criterion in terms of organization design is presented.

6.2.4 Graph Partitioning

Graph partitioning is a discipline devoted to the problem of partition of the set of nodes of a graph whose nodes and/or edges are weighted by non-negative reals. The partition must be balanced (the classes are to be of predefined [usually equal] weight or size when the nodes are of constant weight) and minimize the total weight of between class communications (edges). This is important, for instance, for parallel computations (making equal load per processor while minimizing interprocessor communications) and in very large scale integrated (VLSI) circuits layout (the nodes correspond to chips and edges to wires connecting them).

Since the total weight of between node connections is constant, minimizing between class connections is equivalent to maximizing within class connections as expressed by the uniform partition criterion (6.15) (with zero threshold).

The NP-complete problem of graph bisection (splitting a graph into two equal size parts) has received most attention in graph partitioning. Besides the Lin-Kernighan exchange procedure discussed above, there are three kinds of heuristics which attracted most efforts: (a) min-cut extensions, (b) bisection by a vertex separator, (c) spectral partitioning. Let us describe them in turn.

As it is well known, the Ford-Fulkerson max-flow algorithm splits the graph in the optimal way. However, the sizes of two split parts may be far from equal. This can be corrected with heuristics based on the exchange procedure (Krishnamurty (1984)). Recently, this has become accompanied with an option of replicating some nodes (to have them in both of the parts) so that more interconnections are reduced (Hwang (1995), Liu et al. (1995)).

A vertex separator of a graph (I, A) (A is the matrix of edge weights, $a_{ij} \geq 0$) is such a subset $V \subset I$ that removing it along with all incident edges from the graph results in two disconnected subgraphs of (almost) equal sizes. An existence theorem (Miller et al. (1993) extending Lipton and Tarjan (1979)) says that if G is an (α, k) overlap graph in n dimensions with q nodes, then there exists a vertex separator whose cardinality is at most $O(\alpha k^{1/n} q^{(n-1)/n})$ nodes while each of the disconnected parts has at most $q(n+1)/(n+2)$ nodes. This result underlies algorithms for finding vertex separators by projecting nodes onto lines or spheres so that the separator corresponds to points projected into the midst (see Miller et al. (1993), Demmel (1996) and references therein).

Spectral partitioning is applied to an ordinary graph (the edge weights are zeros or unities) based on that idea that the eigenvector of the data matrix corresponding to its minimal positive eigenvalue corresponds to that direction in which the data "cloud" is most elongated. If we can find this direction, the problem is solved by just cutting it by half. This idea was elaborated by Pothen, Simon and Liou (1990) based on earlier results by Fiedler (1975). Let the number of edges be M. The basic construction is the $N \times M$ incidence matrix $In(G)$ whose columns correspond to the edges having all zero entries except for two incident row-vertices: one is 1, the other is -1. The Laplacian $N \times N$ matrix, $L(G) = In(G)In(G)^T$, has its diagonal entries, (i, i), equal to the number of incident edges, and non-diagonal entries, (i, j), equal to -1 (if (i, i) is an edge) or 0 ((i, j) is not an edge). It is the minimal positive eigenvalue of $L(G)$ and corresponding eigenvector, x, which determine the sought bisection: all entities $i \in I$ with $x_i > 0$ go into one part while the entities with $x_i < 0$ go into the other. Computation of the x (approximately) can be done cost-effectively if graph G is sparse (see also Hagen and Kahng (1992), Demmel (1996)) .

The bisections found can be improved with an exchange (Kernighan-Lin) heuristics. Applying bisections iteratively to a sequence of "coarser" graphs (found by aggregating the graph adjacency matrix around vertices belonging to locally maximal matchings) can make computations faster (see Demmel (1996), Agrawal et al. (1995)).

6.3 Overlapping Clusters

The problem of revealing overlapping clusters with no predefined overlap structure can be put in the approximation clustering framework. In the additive cluster model (6.16), the clusters may overlap (as well as in the approximation model (6.10)). The clustering strategies developed so far exploit the additivity of the model which allows processing one cluster at a time. The strategies thus involve two nested cycles: (a) a major one dealing with sequential preparing data for revealing (updating) a cluster, and (b) a minor one dealing with finding (updating) a single cluster. A straightforward implementation of this idea is in the method of sequential fitting SEFIT (Mirkin (1987, 1990)) which can be applied to any approximation multiclustering model. The method will be explained here only for the additive clustering model:

(a) a residual (similarity) data matrix is obtained by transformation $a_{ij} \leftarrow a_{ij} - \lambda s_i s_j$ where λ and s_i are the intensity and membership of the cluster found at preceding step (all $s_i = 1$ and λ equal to grand mean are taken at the first step);

(b) a cluster is found by minimizing a corresponding single cluster clustering criterion, (5.12) in this case,

$$L^2 = \sum_{i,j \in I} (a_{ij} - \lambda s_i s_j)^2$$

with regard to unknown real (positive) λ and Boolean s_i, $i \in I$. This can be done with a single cluster clustering method;

(c) a stopping criterion is applied (based on a prespecified threshold: the number of clusters or the explained contribution to the data scatter). If No, go to (a); if Yes, end.

The decomposition (6.17) holds with this method, which allows to evaluate the contributions, thus saliences, of different clusters in the data scatter. *Example.* Applying the sequential fitting method for partitioning ten styled digits by the Confusion data considered as a similarity matrix (the diagonal entries removed, the matrix symmetrized, and the grand mean subtracted), we find a cluster sequence presented in Table 11.

□

A doubly alternating minimization technique is employed by Chaturvedi and Carroll (1994) within the same strategy. A somewhat wider approach is suggested in Hartigan (1972).

7 Hierarchical Structure Clustering

7.1 Approximating Binary Hierarchies

A hierarchy S_W is called binary if every nonsingleton cluster $S_w \in S_W$ has exactly two children, $S_w = S_{w1} \cup S_{w2}$, $S_{w1}, S_{w2} \in S_W$. Such a cluster can

Table 11: The SEFIT clusters for 10 digits (Confusion data).

Cluster	Entities	Intensity	Contribution, %
1	1-7	131.04	25.4
2	3-9	98.04	14.2
3	6-8-0	54.71	13.3
4	5-9	70.54	7.4
5	5-6	54.54	4.4
6	1-4	52.04	4.0
7	8-9-0	24.31	2.6

be assigned with a ternary nest indicator, $\phi_w(i)$, which is equal to a_w for $i \in S_{w1}$ and to b_w for $i \in S_{w2}$ and to 0 for $i \notin S_w$. The values of a_w and b_w are chosen so that ϕ_w is centered and normed:

$$a_w = \sqrt{\frac{n_{w2}}{n_{w1}n_w}}, \text{ and } b_w = -\sqrt{\frac{n_{w1}}{n_{w2}n_w}} \tag{7.1}$$

where n_w, n_{w1}, and n_{w2} are cardinalities of S_w and its two children, S_{w1} and S_{w2}, respectively.

It turns out, vectors ϕ_w are mutually orthogonal, which makes the set $\Phi_W = \{\phi_w\}$ an orthonormal basis of all N-dimensional centered vectors. Any data matrix, Y (preliminarily column-centered), can thus be decomposed by the basis:

$$Y = \Phi C \tag{7.2}$$

where $\Phi = (\phi_{iw})$ is the $N \times (N-1)$ matrix of the values of the nest indicators $\phi_w(i)$ and $C = (c_{wk})$ is an $(N-1) \times n$ matrix. The number $N-1$ is the number of nonsingleton clusters in any binary hierarchy. This implies that $C = \Phi^T Y$, that is,

$$c_{wk} = \sum_{i \in I} \phi_{iw} y_{ik} = \sqrt{\frac{n_{w1}n_{w2}}{n_w}}(y_{w1k} - y_{w2k}), \tag{7.3}$$

where y_{w1k} and y_{w2k} are the averages of k-th variable in S_{w1} and S_{w2}, respectively.

An incomplete binary hierarchy which does not satisfy (a) $I \in S_W$ or/and (b) every singleton is in S_W, will be called a cluster hierarchy, divisive if (a) holds, or agglomerative if (b) holds. Corresponding nest indicators still form a basis, though of a space of smaller than $N-1$ dimensionality.

For a cluster hierarchy, S_W, the corresponding bilinear clustering model is similar to that for the single cluster clustering and partitioning:

$$y_{ik} = \sum_{w \in W} c_{wk} \phi_{iw} + e_{ik} \tag{7.4}$$

The square data scatter decomposition here is

$$(Y, Y) = \sum_{w \in W} \mu_w^2 + (E, E) \tag{7.5}$$

where $\mu_w = \phi_w^T Y Y^T \phi_w / \phi_w^T \phi_w$ is an analogue to the singular value concept. It is equal to

$$\mu_w = \sqrt{\frac{n_{w1} n_{w2}}{n_w}} d(y_{w1}, y_{w2}) = \sqrt{\frac{n_{w1} n_w}{n_{w2}}} d(y_{w1}, y_w) \tag{7.6}$$

where y_{w1} and y_{w2} are gravity centers of S_{w1} and S_{w2} and $d(x, y)$ is the Euclidean distance between vectors x, y.

Therefore, finding an optimal (partial) Φ requires maximizing $\sum_{w \in W} \mu_w^2$, the weighted sum of all between-children-center distances. The optimal splitting system probably does not satisfy the minimal distance rule (see p. 39), which makes the problem hard.

However, the sequential fitting greedy strategy can be applied to sequentially maximize one μ_w^2 at a time. For a divisive cluster hierarchy, we may start with $W = \{I\}$ one by one adding optimal splits to it. Criterion to optimize by splitting of S_w into S_{w1} and S_{w2} is

$$\mu_w^2 = \frac{n_{w1} n_{w2}}{n_w} d^2(y_{w1}, y_{w2}) = \frac{n_{w1} n_w}{n_{w2}} d^2(y_{w1}, y_w), \tag{7.7}$$

Maximizing (7.7) is equivalent to finding a two class partition of S_w minimizing the square error clustering criterion D_2 in (6.12). The second formula for μ_w^2 yields that the problem is similar to that in the single principal cluster clustering. The optimal splits satisfy the minimal distance rule which guarantees no more than N^n separations to enumerate. Alternating optimization algorithm also can be applied. Depending on which formula in (7.7) is employed, the starting setting is taken as the two most distant points in S_w (the first formula) or the most "extreme" point in S_w (the second formula).

The first formula for μ_w^2 is exactly the dissimilarity measure appeared in the context of Ward agglomerative clustering (see subsection 6.1.7). Thus, Ward algorithm is exactly the sequential least-squares fitting procedure for model (7.4) starting with the agglomerative cluster hierarchy consisting of singletons only.

Example. The sequential split strategy produced a pattern of splits presented in Fig. 10. The relative cluster value μ_w^2 (which is equal to contribution of the split to the square data scatter) is assigned to each of the three initial splits. □

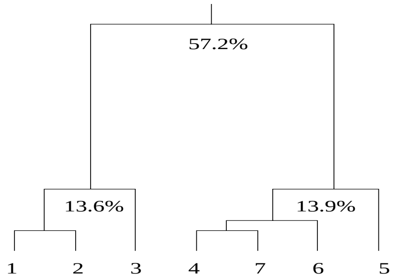

Figure 10: A divisive tree for the data in table 2 found with the least-squares criterion.

7.2 Indexed Hierarchies and Ultrametrics

To draw a hierarchy S_W as a rooted tree, one needs an index function, $h : W \rightarrow R$, assigned to clusters in such a way that $S_{w'} \subset S_w$ implies $h(w') < h(w)$; the value $h(w)$ corresponds to the height of the node w in a drawn representation of S_W (see in Figures 12 (d), 14, and 15).

For any indexed hierarchy, a dissimilarity measure $D = (d(i,j))$ on the set of the entities can be defined as $d(i,j) = h(n[i,j])$ where $n[i,j]$ is the minimum node cluster being the ancestor for both $i,j \in I$. Such a measure is special: it is an ultrametric (the concept introduced by R. Baire in late nineteen century), that is, for any $i,j,k \in I$

$$d(i,j) \leq \min[d(i,j), d(j,k)] \qquad (7.8)$$

Moreover, any ultrametric $D = (d(i,j))$ with the range of the distance values $0 = d_0 < d_1 < ... < d_q$ determines an indexed hierarchy S_D whose clusters S_w with $h(w) = d_\alpha$ are cliques/connected components in the threshold graphs $G_\alpha = \{(i,j) : d(i,j) \leq d_\alpha\}$ $(\alpha = 0, ..., q)$. Every graph G_α is an equivalence relation graph; its connected components are cliques. Thus, the concepts of ultrametric and indexed hierarchy are equivalent. The underlying hierarchy is defined up to any monotone increasing transformation of the ultrametric (index function).

This makes meaningful considering the problem of reconstructing an indexed hierarchy from a dissimilarity matrix in approximation framework. The results found for the problem of approximating a given dissimilarity matrix, (d_{ij}) with an ultrametric, $(d(i,j))$, satisfying inequality $d(i,j) \leq d_{ij}$ are as follows (Johnson (1967), Gower and Ross (1969), Leclerc (1995)). Any spanning tree T on I defines an ultrametric, $d_T(i,j) = \max\{d_{i'j'} : (i',j') \in T(i,j)\}$ where $T(i,j)$ is the unique path between the vertices i and j in tree T. In the case when T is an MST for dissimilarity d, d_T is the maximum ultrametric satisfying inequality $d_T \leq d$, which implies that d_T is an optimal fit according to any criterion monotonously depending on the absolute differences between d and d_T. The clusters of the hierarchy corresponding to d_T (when T is a MST) are connected components of the threshold graphs for original dissimilarity data. Moreover, the hierarchy is that one found with the single linkage method.

The problem of unconstrained approximation of a given dissimilarity by an ultrametric using the least-squares approximation is NP-complete (Day (1987, 1996)) while it is polynomial when L_∞ norm is employed (Agarwala et al. (1995)).

7.3 Fitting in Tree Metrics

The between entity distances in an ultrametric are controlled by the corresponding index function which may seem too restrictive in some substantive problems. A concept of tree metric as a less restrictive clustering structure has emerged. For a tree T on I whose edges are weighted by a positive weight function, w_{ij}, let us define a metric d_{Tw} by the equation $d_{Tw} = \sum_{(i',j') \in T(i,j)} w_{ij}$ where $T(i,j)$ is the only path between i and j in T. It appears (Zaretsky (1965), Buneman (1971)) a metric, (d_{ij}), is a tree metric if and only if the following so-called *four-point condition* is satisfied for every $i, j, k, l \in I$:

$$d_{ij} + d_{kl} \leq \max(d_{ik} + d_{jl}, d_{il} + d_{jk}). \tag{7.9}$$

The proof, actually, can be easily derived from the picture in Fig.11 presenting the general pattern of the tree paths pair-wisely joining the four vertices involved in (7.9). The tree metrics are related to ultrametrics via the so called FKE-transform (Farris, Kluge and Eckart (1970)): let us pick an arbitrary $c \in I$ and define yet another distance on $I - \{c\}$

$$d_c(i,j) = MM + d_{ij} - d_{ic} - d_{jc} \tag{7.10}$$

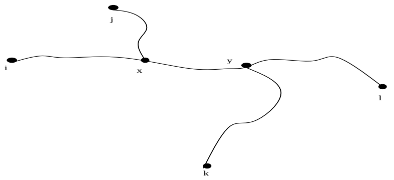

Figure 11: Four-point pattern in an edge weighted tree.

where $MM > 0$ is chosen to make all the values of d_c non-negative. It appears that the following properties are equivalent:

(1) d is a tree metric;

(2) d_c is an ultrametric for any $c \in I$;

(3) d_c is an ultrametric for some $c \in I$.

In Fig.12 an example is presented to show correspondence between a weighted tree and its FKE-transform.

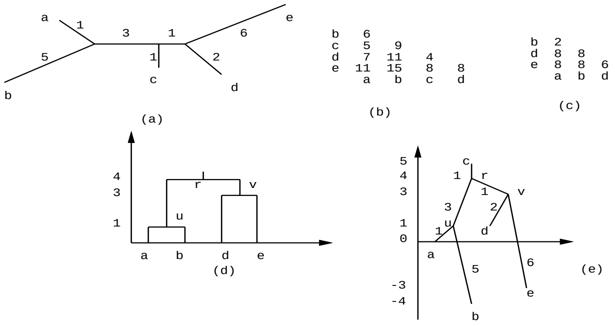

Figure 12: Weighted tree (a) and its tree metric (b) transformed into ultrametric (c) and indexed hierarchy (d) with FKE-transformation; (e) is the tree resulting with the inverse transformation.

There is not much known about problems of approximation of a dissimilarity by tree metrics. However, there are a handful of algorithms for constructing a convenient tree by a dissimilarity data in such a way that if the dissimilarity is a tree metric, the algorithm recovers the corresponding tree. Let us formulate the following algorithm of this kind (Sattah and Tversky (1977), Studier and Keppler (1988)).

Neighbor Joining Algorithm

1. Pick a pair of immediate neighbors, i and j.

2. Form a new node u with its distances $d_{u,k} = (d_{ik} + d_{jk} - d_{ij})/2$, $k \in I - \{i, j\}$, put it in I and remove i and j (after deleting i and j, u becomes a leaf).

3. If there are still some entities of I unremoved, go to step 1 (with the data reduced); otherwise end.

To find a pair of immediate neighbors, the following centrality index can be employed, $c(i,j) = \sum_{k \in I} d(k, T(i,j))$, where $d(k, T(i,j)$ is the tree distance from $k \in I$ to the path $T(i,j)$ between $i, j \in I$ in an underlying tree T. The index can be calculated when T is unknown by using the following formula:

$$c(i,j) = (d_{i+} + d_{j+} - (N-2)d_{ij})/2$$

where $d_{i+} = \sum_{k \in I} d_{ik}$ for any $i \in I$ (see Mirkin (1996), Mirkin et al. [Eds.] (1997) for further detail and references).

Further extensions of the hierarchic cluster structures, first of all, are in the concepts of Robinson matrix and weak hierarchies (see Diday (1986), Hubert and Arabie (1994), Bandelt and Dress (1989, 1992); a review can be found in Mirkin (1996)).

8 Clustering for Aggregable Data

8.1 Box Clustering

We refer to a data matrix $P = (p_{ij})$, $i \in I$, $j \in J$, as an aggregable one if it makes sense to add the entries up to their total, $p_{++} = \sum_{i \in I} \sum_{j \in J} p_{ij}$, as it takes place for contingency, flow or mobility data.

There can be two different goals for the aggregable data analysis: 1) analysis within row or column set similarities, 2) analysis between row and column set interrelations.

To analyze row/column interrelations, a cluster structure called box clustering should be utilized. Two subsets, $V \subseteq I$ and $W \subseteq J$, and a real, μ, represent a box cluster as presented with $|I| \times |J|$ matrix having its entries equal to $\lambda v_i w_j$ where v and w are Boolean indicators of the subsets V and W, respectively.

For the aggregable data, a specific approximation clustering strategy emerges based on the following two features:

(1) it is transformed data entries, $q_{ij} = p_{ij}/p_{i+}p_{+j} - 1$, are to be approximated rather than the original data p_{ij};

(2) it is a weighted least-squares criterion employed rather than the common unweighted one (see Mirkin (1996)).

In the latter reference, the following *box clustering* model is considered. The model is a bilinear equation,

$$q_{ij} = \sum_{t=1}^{m} \mu_t v_{it} w_{jt} + e_{ij} \tag{8.1}$$

to be fit by minimizing

$$L^2 = \sum_{i \in I} \sum_{j \in J} p_{i+}p_{+j}(q_{ij} - \sum_{t=1}^{m} \mu_t v_{it} w_{jt})^2 \tag{8.2}$$

with regard to real μ_t and Boolean v_{it}, w_{jt}, $t = 1, ..., m$.

The following rationales can be suggested to support the box clustering model:

(1) It is a clustering extension of the method of correspondence analysis (widely acknowledged to be a genuine method in analysis and visualization of contingency data, see, for instance, Lebart, Morineau, and Piron (1995)).

(2) When a box $\mu_t v_t w_t^T$ is orthogonal to the other boxes, the optimal μ_t is also a flow index applied, this time, to subsets V_t and W_t:

$$\mu_t = q_{V_t W_t} = (p_{V_t W_t} - p_{V_t +} p_{+W_t})/p_{V_t +} p_{+W_t} \tag{8.3}$$

where $p_{V_t W_t} = \sum_{i \in V_t} \sum_{j \in W_t} p_{ij}$.

(3) The box clusters found with a doubly-greedy SEFIT-based strategy (box clusters are extracted one-by-one, and each box cluster is formed with sequential adding/removing a row/column entity) represent quite deviant fragments of the data table (Mirkin (1996)).

The problem of finding of an optimal box, at a single SEFIT step, by maximizing

$$\frac{(\sum_{i \in V} \sum_{j \in W} p_{i+} p_{+j} q_{ij})^2}{\sum_{i \in V} p_{i+} \sum_{j \in J} p_{+j}} \tag{8.4}$$

over $V \subseteq I$ and $W \subseteq J$, seems to be a combinatorial problem deserving further investigation.

Example. Applied to the Worries data in Table 7, the aggregable box clustering algorithm produces 6 clusters; the total contribution of the clusters in the initial value Φ^2 equals some 90 % (see table 12).

Table 12: Box cluster structure of the Worries data set.

Box	Columns	Rows	RCP, %	Contrib., %
1	ASAF, IFAA	PER	79.5	34.5
2	EUAM, IFEA	PER	-46.0	20.8
3	ASAF, IFAA	POL, ECO	-40.5	9.9
4	IFEA	OTH, POL	46.1	9.7
5	EUAM	POL, MIL, ECO, MTO	18.5	9.3
6	IFEA, ASAF, IFAA, IFI	MIL, MTO	-17.5	5.5

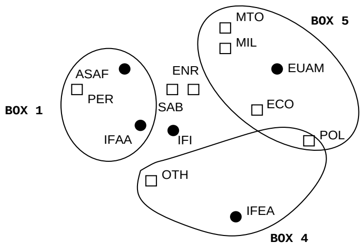

Figure 13: Positive RCP boxes in the correspondence analysis factor plane.

The content of Table 12 corresponds to the traditional joint display given by the first two correspondence analysis factors (see Fig.13 where the columns and the rows are presented by the circles and the squares, respectively).

Due to the model's properties, all the boxes with positive aggregate flow index (RCP) values (clusters 1, 4, and 5) correspond to the continuous fragments of the display (shown on Fig.13); boxes with the negative RCP values are associated with distant parts of the picture. □

Box clustering problems can arise for other data types. Levit (1988) provides a simple (matching based) solution to the problem of finding an all unity box of maximum perimeter in a Boolean data table.

8.2 Bipartitioning

We refer to a box clustering problem as that of bipartitioning when the boxes are generated by partitions on each of the sets, I and J. Let $S = \{V_t\}$

be a partition of I, and $T = \{W_u\}$, of J, so that every pair (t, u) labels corresponding box (V_t, W_u) and its weight μ_{tu}. In corresponding specification of the model (8.1)-(8.2), the optimal values μ_{tu} are $q_{V_tW_u}$ in (8.3).

Due to mutual orthogonality of the boxes (V_t, W_u), a decomposition of the weighted squared scatter of the data, q_{ij}, onto the minimized criterion L^2 (8.2) and the bipartition part which is just the sum of terms having format of (8.4), can be made analogously to those in (6.17). The optimization problem here is an analogue of that related to (6.17). An equivalent reformulation of the problem involves aggregation of the data based on the Pearson contingency coefficient. Let us aggregate the $|I| \times |J|$ table $P = (p_{ij})$ into $|S| \times |T|$ table $P(S,T) = (p_{tu})$ where $p_{tu} = \sum_{i \in V_t} \sum_{j \in W_u} p_{ij}$. In this notation, the original table is just $P = P(I, J)$. Then, the contingency coefficient is

$$X^2(S,T) = \sum_{t,u} \frac{(p_{tu} - p_{t+}p_{+u})^2}{p_{t+}p_{+u}}.$$

It is not difficult to see, that the data scatter decomposition, due to the bilinear model under consideration, is nothing but

$$X^2(I, J) = X^2(S,T) + L^2 \qquad (8.5)$$

which means that the bipartitioning problem is equivalent to that of finding such an aggregate $P(S,T)$ which maximizes $X^2(S,T)$.

Alternating and agglomerating optimization clustering procedures can be easily extended to this case (Mirkin (1996)). Reformulated in geometric clustering terms, they involve the chi-squared distance defined in section 2.

8.3 Aggregation of Flow Tables

The flow table is an aggregable data table $P = (p_{ij})$ where $I = J$ as, for instance, in brand switching or digit confusion or input-output tables. Aggregation problem for such a table can be stated as that of bipartitioning with coinciding partitions, $S = T$, or equivalently, of finding an aggregate table $P(S, S)$ maximizing corresponding Pearson contingency coefficient $X^2(S, S)$. Another formulation involves finding such a partition, $S = \{V_1, ..., V_m\}$, that the aggregate flow index values, q_{tu}, satisfy equations

$$q_{ij} = q_{tu} + \epsilon_{ij}, \ i \in V_t, \ j \in V_u \qquad (8.6)$$

and minimize the criterion, $\sum_{t,u} \sum_{i \in V_t} \sum_{j \in V_u} p_{i+}p_{+j}(q_{ij} - q_{tu})^2$.

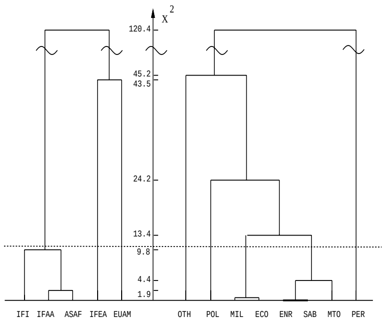

Figure 14: Hierarchical biclustering results for the Worries data.

Applying the agglomerative clustering algorithm (by minimizing decrement of $X^2(S,S)$ at each agglomeration step) to Confusion data table (all the entries taken into account), we obtain the hierarchy presented in Fig. 15. The hierarchy is indexed by the level of unexplained X^2 at each level of aggregation.

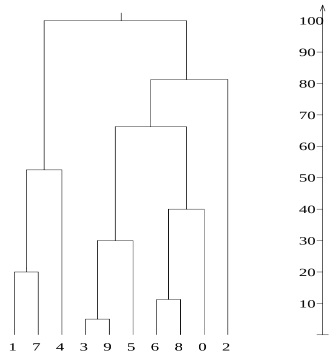

Figure 15: Results of agglomerative chi-square based aggregation for Confusion data.

9 Conclusion

Historically, clustering has appeared mostly as a set of ad hoc techniques such as K-Means and Lance-Williams hierarchical clustering algorithms.

This is why finding appropriate optimal problems to substantiate, specify, modify and extend clustering algorithms is an important part of the clustering theory. The clustering techniques are also local, which naturally leads to problems of revealing such classes of criteria that can be globally optimized with local search. The matroids are a well known example of such a class. The concave/convex set functions touched in 5.2 is another class of this kind.

Another issue for theoretical development is analysis of the properties of the optimality criteria and interrelations between them since they usually have no independent meaning except for those in specific industrial or computational applications. As we tried to demonstrate, there is an intrinsic similarity between many clustering techniques traditionally viewed different but being, in fact, different local search techniques for the same criterion.

There is a two-way interconnection between combinatorial optimization and clustering. The combinatorial theory gives to clustering well established concepts and methods while clustering pays back by supplying a stock of relevant problems and heuristical computational techniques that are computationally efficient in solving hard combinatorial optimization problems. It seems, every polynomially solved combinatorial problem (not only min-cut or maximum-density-subgraph, but also matching, assignment, etc.) can contribute to clustering. On the other hand, it should be expected that the cluster-based search and optimization techniques for combinatorial problems, already under testing, will be expanding when larger sizes of data will be processed. There are a few concepts such as minimum spanning tree or greedy algorithms that have a good theoretical standing in both of the fields. We also can see an imbalance in the support provided by the combinatorial optimization theory to optimal clustering problems: the similarity-based (graph-theoretic) constructions are much better explored than those coordinate-based. In particular, the problem of estimating of efficacy of the alternating minimization partitioning algorithm (K-Means) seems a good subject for potential analysis: the only estimate known, $N^{|}V|$, exploits only one feature, linear separability, of the clusters, while there are more to take into account. Though, some work on combinatorial analysis in coordinate spaces is being also done (see, for instance, Callahan and Kosaraju (1995), Edelsbrunner (1987), Yao (1982)).

Among the other topics of interest are the following:

(a) theoretical and computational support of greedily optimized classes of set functions such as those mentioned above (sub-modular, concave/convex) and corresponding single and other cluster structures;

(b) interconnections between the traditional eigen/singular value decompositions of matrices and those restricted by discrete cluster structures (as the nest or partition indicator bases);

(c) further exploring heuristics for hard clustering problems (see, for instance, Guénoche, Hansen, and Jaumard (1991), Hansen, Jaumard, and Mladenovic (1995), Hsu and Nemhauser (1979), Johnson and Trick (1996), McGuinness (1994), Pardalos, Rendl, Wolkowicz (1994)),

(d) developing mathematical theories for clustering and window-based clustering in spatial data sets;

(e) finding other discrete clustering structures (set systems) and related problems as emerging in application areas.

In general, the optimization clustering problems have extensive overlaps with those in the semidefinite programming (Vandenberghe and Boyd (1996), Pardalos and Wolkowicz (1997)) and the quadratic assignment (Hubert (1987), Pardalos, Rendl, Wolkowicz (1994)). There has not been much done in applying these global optimization techniques to clustering problems.

References

[1] R. Agarwala, V. Bafna, M. Farach, B. Narayanan, M. Paterson, and M. Thorup, On the approximability of numerical taxonomy, (DIMACS Technical Report 95-46, 1995).

[2] A. Agrawal and P. Klein, Cutting down on fill using nested dissection: Provably good elimination orderings, in A. George, J.R. Gilbert, and J.W.H. Liu (eds.) *Sparse Matrix Computation*, (London, Springer-Verlag, 1993).

[3] P. Arabie, S.A. Boorman, and P.R. Levitt, Constructing block models: how and why, *Journal of Mathematical Psychology* Vol.17 (1978) pp. 21-63.

[4] P. Arabie and L. Hubert, Combinatorial data analysis, *Annu. Rev. Psychol.* Vol. 43 (1992) pp. 169-203.

[5] P. Arabie, L. Hubert, G. De Soete (eds.) *Classification and Clustering*, (River Edge, NJ: World Scientific Publishers, 1996).

[6] C. Arcelli and G. Sanniti di Baja, Skeletons of planar patterns, in T.Y. Kong and A. Rosenfeld (eds.) *Topological Algorithms for Digital Image Processing*, (Amsterdam, Elsevier, 1996) pp. 99-143.

[7] H.-J. Bandelt and A.W.M. Dress, Weak hierarchies associated with similarity measures – an additive clustering technique, *Bulletin of Mathematical Biology* Vol. 51 (1989) pp. 133-166.

[8] H.-J. Bandelt and A.W.M. Dress, A canonical decomposition theory for metrics on a finite set, *Advances of Mathematics* Vol. 92 (1992) pp. 47-105.

[9] J.-P. Benzécri (1973) *L'Analyse des Données*, (Paris, Dunod, 1973).

[10] P.Brucker (1978) On the complexity of clustering problems, in R.Henn et al. (eds.) *Optimization and Operations Research*, (Berlin, Springer, 1978) pp. 45 - 54.

[11] P. Buneman, The recovery of trees from measures of dissimilarity, in F. Hodson, D. Kendall, and P. Tautu (eds.) *Mathematics in Archaeological and Historical Sciences*, (Edinburgh, Edinburgh University Press, 1971) pp. 387-395.

[12] P.B. Callahan and S.R. Kosaraju, A decomposition of multidimensional point sets with applications to k-nearest neighbors and n-body potential fields, *Journal of ACM* Vol. 42 (1995) pp. 67-90.

[13] A. Chaturvedi and J.D. Carroll, An alternating optimization approach to fitting INDCLUS and generalized INDCLUS models, *Journal of Classification* Vol. 11 (1994) pp. 155-170.

[14] P.Crescenzi and V.Kann, *A compendium of NP optimization problems*, (URL site: http://www.nada.kth.se/ṽiggo/problemlist/ compendium2, 1995).

[15] W.H.E. Day, Computational complexity of inferring phylogenies from dissimilarity matrices, *Bulletin of Mathematical Biology* Vol. 49 (1987) pp. 461-467.

[16] W.H.E. Day (1996) Complexity theory: An introduction for practitioners of classification, In: P. Arabie, L.J. Hubert, and G. De Soete (Eds.) *Clustering and Classification*, World Scientific: River Edge, NJ, 199-233.

[17] M. Delattre and P. Hansen, Bicriterion cluster analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* Vol. 4 (1980) pp. 277-291.

[18] J. Demmel, *Applications of Parallel Computers*, (Lectures posted at web site: http://HTTP.CS.Berkeley.EDU/ demmel/cs267/, 1996).

[19] E. Diday, Orders and overlapping clusters by pyramids, in J. de Leeuw, W. Heiser, J. Meulman, and F. Critchley (eds.) *Multidimensional Data Analysis*, (Leiden, DSWO Press, 1986) pp. 201-234.

[20] A.A. Dorofeyuk, Methods for automatic classification: A Review, *Automation and Remote Control* Vol.32 No.12 (1971) pp. 1928-1958.

[21] A.W.M. Dress and W. Terhalle, Well-layered maps - a class of greedily optimizable set functions, *Appl. Math. Lett.* Vol.8 No.5 (1995) pp. 77-80.

[22] H. Edelsbrunner, *Algorithms in Combinatorial Geometry* (New York, Springer Verlag, 1987).

[23] M. Fiedler, A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory, *Czech. Math. Journal* Vol.25 (1975) pp. 619–637.

[24] D.W. Fisher, Knowledge acquisition via incremental conceptual clustering, *Machine Learning* Vol.2 (1987) pp. 139-172.

[25] K. Florek, J. Lukaszewicz, H. Perkal, H. Steinhaus, and S. Zubrzycki, Sur la liason et la division des points d'un ensemble fini, *Colloquium Mathematicum* Vol.2 (1951) pp. 282-285.

[26] G. Gallo, M.D. Grigoriadis, and R.E. Tarjan, A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing* Vol.18 (1989) pp. 30-55.

[27] M.R. Garey and D.S. Johnson, *Computers and Intractability: a guide to the theory of NP-completeness*, (San Francisco, W.H.Freeman and Company, 1979).

[28] M. Gondran and M. Minoux, *Graphs and Algorithms*, (New-York, J.Wiley & Sons, 1984).

[29] J.C. Gower and G.J.S. Ross, Minimum spanning tree and single linkage cluster analysis, *Applied Statistics* Vol.18 pp. 54-64.

[30] D. Gusfield, Efficient algorithms for inferring evolutionary trees, *Networks* Vol.21 (1991) pp. 19-28.

[31] A. Guénoche, P. Hansen, and B. Jaumard, Efficient algorithms for divisive hierarchical clustering with the diameter criterion, *Journal of Classification* Vol.8 (1991) pp. 5-30.

[32] L. Hagen, A.B. Kahng, New spectral methods for ratio cat partitioning and clustering, *IEEE Transactions on Computer-Aided Design* Vol.11 No.9 (1992) pp. 1074-1085.

[33] P. Hansen, B. Jaumard, and N. Mladenovic, How to choose K entities among N. in I.J. Cox, P. Hansen, and B. Julesz (eds.) *Partitioning Data Sets*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Providence, American Mathematical Society, 1995) pp. 105-116.

[34] J.A. Hartigan, Direct clustering of a data matrix, *Journal of American Statistical Association* Vol. 67 (1972) pp. 123–129.

[35] J.A. Hartigan, *Clustering Algorithms*, (New York, J.Wiley & Sons, 1975).

[36] W.-L. Hsu and G.L. Nemhauser, Easy and hard bottleneck location problems, *Discrete Applied Mathematics* Vol.1 (1979) pp. 209-215.

[37] L.J. Hubert, *Assignment Methods in Combinatorial Data Analysis*, (New York, M. Dekker, 1987).

[38] L. Hubert and P. Arabie, The analysis of proximity matrices through sums of matrices having (anti)-Robinson forms, *British Journal of Mathematical and Statistical Psychology* Vol.47 (1994) pp. 1-40.

[39] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, (Englewood Cliffs, NJ, Prentice Hall, 1988).

[40] K. Janich, *Linear Algebra*, (New York, Springer-Verlag, 1994).

[41] D.S. Johnson and M.A. Trick (eds.), *Cliques, Coloring, and Satisfiability*. DIMACS Series in Discrete mathematics and theoretical computer science, V.26. (Providence, RI, AMS, 1996) 657 p.

[42] S.C. Johnson, Hierarchical clustering schemes, *Psychometrika* Vol.32 (1967) pp. 241-245.

[43] Y. Kempner, B. Mirkin, and I. Muchnik, Monotone linkage clustering and quasi-concave set functions. *Applied Mathematics Letters* Vol.10 No.4 (1997)pp. 19-24.

[44] G. Keren and S. Baggen, Recognition models of alphanumeric characters, *Perception and Psychophysics* (1981) pp. 234-246.

[45] B. Kernighan and S. Lin, An effective heuristic procedure for partitioning of electrical circuits, *The Bell System Technical Journal* Vol.49 No.2 (1970) pp. 291-307.

[46] B. Krishnamurthy, An improved min-cut algorithm for partitioning VLSI networks, *IEEE Transactions on Computers* Vol.C-33 No.5 (1984) pp. 438-446.

[47] V. Kupershtoh, B. Mirkin, and V. Trofimov, Sum of within partition similarities as a clustering criterion, *Automation and Remote Control* Vol.37 No.2 (1976) pp. 548-553.

[48] V. Kupershtoh and V. Trofimov, An algorithm for analysis of the structure in a proximity matrix, *Automation and Remote Control* Vol.36 No.11 (1975) pp. 1906-1916.

[49] G.N. Lance and W.T. Williams, A general theory of classificatory sorting strategies: 1. Hierarchical Systems, *Comp. Journal* Vol.9 (1967) pp. 373-380.

[50] L. Lebart, A. Morineau, and M. Piron, *Statistique Exploratoire Multidimensionnelle*, (Paris, Dunod, 1995).

[51] B. Leclerc, Minimum spanning trees for tree metrics: abridgments and adjustments, *Journal of Classification* Vol.12 (1995) pp. 207-242.

[52] V. Levit, An algorithm for finding a maximum perimeter submatrix containing only unity, in a zero/one matrix, in V.S. Pereverzev-Orlov (ed.) *Systems for Transmission and Processing of Data*, (Moscow, Institute of Information Transmission Science Press, 1988) pp. 42-45 (in Russian).

[53] L. Libkin, I. Muchnik, and L. Shvarzer, Quasi-linear monotone systems, *Automation and Remote Control* Vol.50 pp. 1249-1259.

[54] R.J. Lipton and R.E. Tarjan, A separator theorem for planar graphs, *SIAM Journal of Appl.Math.* Vol.36 (1979) pp. 177-189.

[55] S. McGuinness, The greedy clique decomposition of a graph, *Journal of Graph Theory* Vol.18 (1994) pp. 427-430.

[56] G.L.Miller, S.-H. Teng, W.Thurston, and S.A.Vavasis, Automatic mesh partitioning, in A. George, J.R. Gilbert, and J.W.H. Liu (eds.) *Sparse Matrix Computations: Graph Theory Issues and Algorithms*, (London, Springer-Verlag, 1993).

[57] G.W. Milligan, A Monte Carlo study of thirty internal criterion measures for cluster analysis, *Psychometrika* Vol.46 (1981) pp. 187-199.

[58] B. Mirkin, Additive clustering and qualitative factor analysis methods for similarity matrices, *Journal of Classification* Vol.4 (1987) pp. 7-31; Erratum Vol.6 (1989) pp. 271-272.

[59] B. Mirkin, A sequential fitting procedure for linear data analysis models, *Journal of Classification* Vol.7 (1990) pp. 167-195.

[60] B. Mirkin, Approximation of association data by structures and clusters, in P.M. Pardalos and H. Wolkowicz (eds.) *Quadratic Assignment and Related Problems.* DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (Providence, American Mathematical Society, 1994) pp. 293-316.

[61] B. Mirkin, *Mathematical Classification and Clustering*, (Dordrecht-Boston-London, Kluwer Academic Publishers, 1996).

[62] B. Mirkin, F. McMorris, F. Roberts, A. Rzhetsky (eds.) *Mathematical Hierarchies and Biology.* DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (Providence, RI, AMS, 1997) 389 p.

[63] I. Muchnik and V. Kamensky, MONOSEL: a SAS macro for model selection in linear regression analysis, in *Proceedings of the Eighteenth Annual SAS* Users Group International Conference* (Cary, NC, SAS INstitute Inc., 1993) pp. 1103-1108.

[64] I.B. Muchnik and L.V. Schwarzer, Nuclei of monotone systems on set semilattices, *Automation and Remote Control* Vol.52 (1989) 1993) pp. 1095-1102.

[65] I.B. Muchnik and L.V. Schwarzer, Maximization of generalized characteristics of functions of monotone systems, *Automation and Remote Control* Vol.53 (1990) pp. 1562-1572.

[66] J. Mullat, Extremal subsystems of monotone systems: I, II; *Automation and Remote Control* Vol.37 (1976) pp. 758-766, pp. 1286-1294.

[67] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, (Englewood Cliffs, NJ, Prentice–Hall, 1982).

[68] P.M. Pardalos, F. Rendl, and H. Wolkowicz, The quadratic assignment problem: a survey and recent developments. in P. Pardalos and H. Wolkowicz (eds.) *Quadratic Assignment and Related Problems*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, v. 16. (Providence, American Mathematical Society, 1994).

[69] Panos M. Pardalos and Henry Wolkowicz (Eds.) *Topics in Semidefinite and Interior-Point Methods*. Fields Institute Communications Series (Providence, American Mathematical Society, 1997).

[70] A. Pothen, H.D. Simon, K.-P. Liou, Partitioning sparse matrices with eigenvectors of graphs, *SIAM Journal on Matrix Analysis and Applications* Vol.11 (1990) pp. 430-452.

[71] S. Sattah and A. Tversky, Additive similarity trees, *Psychometrika* Vol.42 (1977) pp. 319-345.

[72] J. Setubal and J. Meidanis, *Introduction to Computational Molecular Biology*, (Boston, PWS Publishing Company, 1997).

[73] R.N. Shepard and P. Arabie, Additive clustering: representation of similarities as combinations of overlapping properties, *Psychological Review* Vol.86 (1979) pp. 87-123.

[74] J.A. Studier and K.J. Keppler, A note on neighbor-joining algorithm of Saitou and Nei, *Molecular Biology and Evolution* Vol.5 (1988) pp. 729-731.

[75] L. Vandenberghe and S. Boyd, Semidefinite programming, *SIAM Review* Vol. 38 (1996) pp. 49-95.

[76] B. Van Cutsem (Ed.), *Classification and Dissimilarity Analysis*, Lecture Notes in Statistics, 93 (New York, Springer-Verlag, 1994).

[77] J.H. Ward, Jr, Hierarchical grouping to optimize an objective function, *Journal of American Statist. Assoc.* Vol.58 (1963) pp. 236-244.

[78] D.J.A. Welsh, *Matroid Theory*, (London, Academic Press, 1976).

[79] A.C. Yao, On constructing minimum spanning trees in k-dimensional space and related problems, *SIAM J. Comput.* Vol.11 (1982) pp. 721-736.

[80] C.T. Zahn, Approximating symmetric relations by equivalence relations, *J. Soc. Indust. Appl. Math.* Vol. 12, No. 4.

[81] K.A. Zaretsky, Reconstruction of a tree from the distances between its pendant vertices, *Uspekhi Math. Nauk (Russian Mathematical Surveys)* Vol.20 pp. 90-92 (in Russian).