Journal of Classification 6:177-193 (1989)

Maximum Sum-of-Splits Clustering

P. Hansen
Rutgers University

B. Jaumard
Ecole Polytechnique de Montréal

O. Frank
University of Stockholm

Abstract: Consider N entities to be classified, and a matrix of dissimilarities between pairs of them. The split of a cluster is the smallest dissimilarity between an entity of this cluster and an entity outside it. The single-linkage algorithm provides partitions into M clusters for which the smallest split is maximum. We study here the average split of the clusters or, equivalently, the sum of splits. A $\Theta(N^2)$ algorithm is provided to determine maximum sum-of-splits partitions into M clusters for all M between $N-1$ and 2, using the dual graph of the single-linkage dendrogram.

Acknowledgments: The work of the first author was supported in part by AFOSR grant 0271 to Rutgers University and was done in part during a visit to GERAD, Ecole Polytechnique de Montréal, whose support is gratefully acknowledged. The work of the second author was supported by NSERC grant GP0036426 and by FCAR grant 89EQ4144. We are grateful to two anonymous referees for many insightful comments.

Authors' Addresses: P. Hansen, Rutgers Center for Operations Research, Hill Center for the Mathematical Sciences, Rutgers University, New Brunswick, New Jersey 08903, USA, B. Jaumard, GERAD and Département de Mathématiques Appliquées, Ecole Polytechnique, Campus de l'Université de Montréal, Case Postale 6079, Succursale "A", Montréal, Québec, H3C 3A7, Canada, and O. Frank, Department of Statistics, University of Stockholm, S-10691 Stockholm, Sweden.

Résumé: Soient N objets à classifier et une matrice de dissimilarités entre paires de ces objets. L'écart d'une classe est la plus petite dissimilarité entre un objet de cette classe et un objet en dehors d'elle. L'algorithme du lien simple fournit des partitions en M classes dont le plus petit écart est maximum. On étudie l'écart moyen des classes, ou, ce qui est équivalent, la somme des écarts. On propose un algorithme en $\Theta(N^2)$ pour déterminer des partitions en M classes dont la somme des écarts est maximum pour M allant de $N-1$ à 2, basé sur le graphe dual du dendrogramme de la méthode du lien simple.

Keywords: Partition; Split; Dendrogram; Dual graph; Complexity; Polynomial algorithm.

1. Introduction

Cluster analysis (see e.g., Hartigan 1975 and Gordon 1981) aims at finding homogeneous and/or well separated subsets, called clusters, of a given set of N entities. The concepts of homogeneity and separation can be made precise in a variety of ways. It is often assumed that all differences between values of relevant characteristics for any given pair of entities can be summarized by a single number, called dissimilarity. Separation can then be expressed in terms of dissimilarities between entities in a cluster and entities outside of it. A fruitful approach is to focus on the smallest such dissimilarity, as is done in the single-linkage algorithm, and call it the *split* of the cluster. The split of a partition can then be defined as the smallest of its clusters' splits. As shown in Delattre and Hansen (1980), the single-linkage algorithm provides partitions with maximum split at all levels of the hierarchy (see also Zahn 1971, Leclerc 1977, and Hubert 1977 for related results). This method thus optimizes a mathematically well-defined concept of separation. Moreover, as shown below, the single-linkage algorithm also maximizes the sum of the splits of the $2N-1$ clusters appearing in the hierarchy. It does not, however, maximize the sum of the splits of the clusters of each partition of this hierarchy.

The classifier may be more interested in the average split of the clusters of a partition than in the minimum of its splits. Indeed, in some cases, the minimum split may be much smaller than the average split, and thus may be a poor estimate of the separation of all clusters. He will then want to determine the partition with maximum average split or, which is equivalent, the partition into a given number M of clusters with maximum sum of splits. The main result of the present paper is a $\Theta(N^2)$ algorithm to solve that problem for all M between 2 and $N-1$. This is done by determining longest paths with 2 to $N-1$ arcs in the dual graph of the single-linkage dendrogram. The resulting partitions do not necessarily form a hierarchy.

The criteria of the single-linkage method and of the proposed algorithm express in a mathematically precise way two aspects of separation. Other well-known methods of clustering, such as the complete-linkage algorithm or the average-linkage algorithm are more oriented towards aspects of homogeneity. However, to the best of our knowledge, these methods do not optimize any mathematically well-defined criterion, except at each individual iteration. In other words, they are greedy algorithms without an explicit objective function. Homogeneity can be made precise by using the concepts of diameter of a cluster, i.e., maximum dissimilarity between two entities of that cluster, and of diameter of a partition, i.e., maximum of the diameters of the clusters of that partition. Hansen and Delattre (1978) have shown how minimum diameter partitions can be obtained by graph coloring methods. A problem dual to that of this paper is to find minimum sum-of-diameters partitions. Brucker (1978) has shown that it is *NP*-complete for $M \geq 3$ clusters. Hansen and Jaumard (1987) provide an $O(N^3 \log N)$ algorithm for the case $M = 2$. Monma and Suri (1989) recently proposed an $O(N^2)$ algorithm for this last case under the additional assumption that entities are points in the plane and dissimilarities are equal to the Euclidean distances between them.

The present paper is organized as follows. The maximum sum-of-splits clustering problem is formulated in the next Section. Properties of the single-linkage algorithm are discussed in Section 3. Section 4 is devoted to a basic property of maximum sum-of-splits partitions. Our algorithm is presented in Section 5. Maximum split and maximum sum-of-splits partitions for a data set from the literature are compared in Section 6. Conclusions are drawn in the last section.

2. Problem Statement

Let $O = \{O_1, O_2, \ldots, O_N\}$ denote a set of N entities to be classified and $D = (d_{kl})$ a matrix of dissimilarities between all pairs of those entities. As usual, it is assumed that $d_{kl} \geq 0$, $d_{kl} = d_{lk}$ and $d_{kk} = 0$ for $k, l = 1,2,\ldots,N$, but that the triangular inequality $d_{kl} + d_{lm} \geq d_{km}$ need not necessarily hold.

Let $P_M = \{C_1, C_2, \ldots, C_M\}$ denote a partition of O into M clusters; hence $C_j \neq \emptyset$, $C_i \cap C_j = \emptyset$ for $i \neq j$, $i, j = 1,2,\ldots,M$ and $\cup_{j=1,2,\ldots,M} C_j = O$. As stated above, the split $s(C_j)$ of cluster C_j is the smallest dissimilarity between an entity in C_j and an entity not in C_j:

$$s(C_j) = \min_{k,l/O_k \in C_j, O_l \notin C_j} d_{kl}$$

and the split $s(P_M)$ of the partition P_M is:

$$s(P_M) = \min_{j=1,2,\ldots,M} s(C_j) .$$

The sum of splits of P_M's clusters, denoted $ss(P_M)$, is thus:

$$ss(P_M) = \sum_{j=1}^{M} s(C_j) .$$

Any hierarchical clustering algorithm generates a hierarchy H of partitions P_N, $P_{N-1},\ldots,P_1$, i.e., if $C_i \in P_k$, $C_j \in P_l$ and $k > l$ either $C_i \cap C_j = \emptyset$ or $C_i \subset C_j$. Hence any hierarchy H is characterized by $2N-1$ clusters C_j which are pairwise either disjoint or included one in the other. The sum of the splits of H's clusters denoted $ss(H)$, is thus:

$$ss(H) = \sum_{j=1}^{2N-1} s(C_j)$$

where, by convention, $s(C_{2N-1}) = s(O) = 0$.

Let Π_M denote the set of all partitions of O into M non-empty clusters. The *maximum sum-of-splits clustering problem* or *maximum average split clustering problem* (where the optimum value is divided by M) may be formulated:

$$\text{Determine} \quad P_M \in \Pi_M$$

such that $ss(P_M)$ is maximum for $M = 2,3,\ldots,N-1$.

3. Properties of the Single-Linkage Algorithm

Using the concepts of splits of clusters and of partitions, the well-known single-linkage algorithm (SLA) can be expressed as follows:

(a) Let $P_N = \{C_1, C_2, \ldots, C_N\}$ where $C_j = \{O_j\}$ for $j = 1,2,\ldots,N$ and $k = 0$.

(b) Find two clusters C_i and $C_j \in P_{N-k}$ such that $s(C_i) = s(C_j) = s(P_{N-k})$ (and in case of ties in the split values for more than two clusters there exist $O_m \in C_i$, $O_p \in C_j$ such that $d_{mp} = s(C_i)$).

(c) Obtain P_{N-k-1} from P_{N-k} by setting $C_{N+k+1} = C_i \cup C_j$. Set $k \leftarrow k+1$. If $k < N-1$, return to (b).

This algorithm maximizes $s(P_M)$ for all M. We now study its effect on $ss(H)$. Let HI denote the set of all hierarchies of partitions of O.

Theorem 1 *The single-linkage algorithm maximizes for all $H \in HI$ the sum $ss(H)$ of the $2N-1$ splits of the clusters of the hierarchy.*

Proof. Let H^* denote a hierarchy of partitions O with a maximum sum-of-splits $ss(H^*)$. Let d_{kl} denote the minimum dissimilarity between pairs of entities of O. Either $\{O_k,O_l\}$ is a cluster of H^* or not. If not, let us define a modified hierarchy $\hat{H}$ as follows:

(a) For all clusters C_i belonging to H^* such that $O_k \in C_i$ and $O_l \notin C_i$, add O_l to C_i.
(b) For all clusters C_j belonging to H^* such that $O_l \in C_j$ and $O_k \notin C_j$, remove O_l from C_j.
(c) A duplicate class has been formed: in (a) if O_l was first fused with a cluster C_p containing O_k and in (b) if O_l was first fused with a cluster C_q not containing O_k. This duplicate cluster is equal to $C_p \cup \{O_l\}$ in the former case, and to C_q in the latter one; delete it.
(d) Add the cluster $\{O_k,O_l\}$.

Now all clusters which have been modified did contain either O_k or O_l but not both. Their split was therefore minimum and equal to d_{kl}. They are replaced by clusters containing $\{O_k,O_l\}$, the split of which cannot be smaller. Hence $ss(\hat{H}) \geq ss(H^*)$.

In H^*, or $\hat{H}$, one has the cluster $\{O_k,O_l\}$. Entities O_k, O_l can therefore be fused, as in the single-linkage algorithm, and the above argument iterated. This proves the theorem. ∎

Consider now the graph $G = (V,E)$ associated with the dissimilarity matrix D (vertices $v_j \in V$ correspond to entities $O_j \in O$, edges $\{v_k,v_l\} \in E$ are weighted by $d_{kl} \in D$). A cocycle $\omega(A)$ with $A \subset V$ is the set of edges $\{v_k,v_l\} \in E$ such that $v_k \in A$, $v_l \in V \backslash A$. A classical result of graph theory (Rosenstiehl 1967) is that all minimum spanning trees of G contain an edge of minimum weight of all cocycles $\omega(A)$. This implies that the split of any cluster is equal to the weight of an edge of the minimum spanning tree of G.

While the single-linkage algorithm maximizes both $ss(H)$ and $s(P_M)$ for $M = N-1, N-2, \ldots, 2$, it does not maximize the sum-of-splits $ss(P_M)$ of the partitions P_M for all values of M. This is shown in the following example. Consider $O = \{O_1,O_2,O_3,O_4,O_5,O_6,O_7,O_8\}$ and the matrix of dissimilarities given in Table 1. A minimum spanning tree is represented in Figure 1. The single-linkage algorithm gives a partition $P_4 = \{ \{O_1,O_2\}, \{O_3,O_5\}, \{O_4,O_7,O_8\}, \{O_6\} \}$ with $ss(P_4) = 11 + 8 + 11 + 8 = 38$. However, the maximum sum-of-splits partition $\hat{P}_4 = \{ \{O_1\}, \{O_2\}, \{O_3,O_5,O_6\}, \{O_4,O_7,O_8\} \}$ is such that $ss(\hat{P}_4) = 7 + 7 + 17 + 11 = 42$ (see Figure 2).

Table 1. Dissimilarities

	1	2	3	4	5	6	7	8
1	0	7	17	11	18	18	12	13
2	7	0	17	15	20	25	14	11
3	17	17	0	19	3	8	22	20
4	11	15	19	0	19	20	2	4
5	18	20	3	19	0	9	18	20
6	18	25	8	20	9	0	19	18
7	12	14	22	2	18	19	0	7
8	13	11	20	4	20	18	7	0

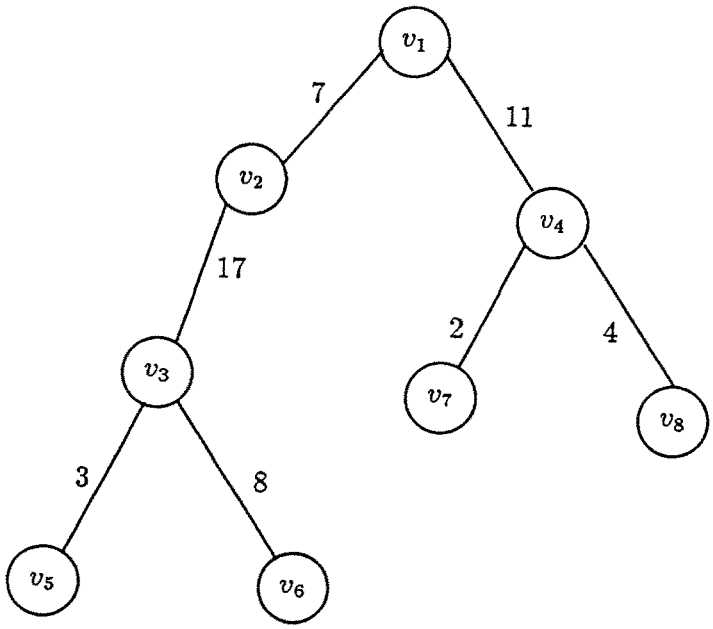

Figure 1. Minimum spanning tree of Example 1.

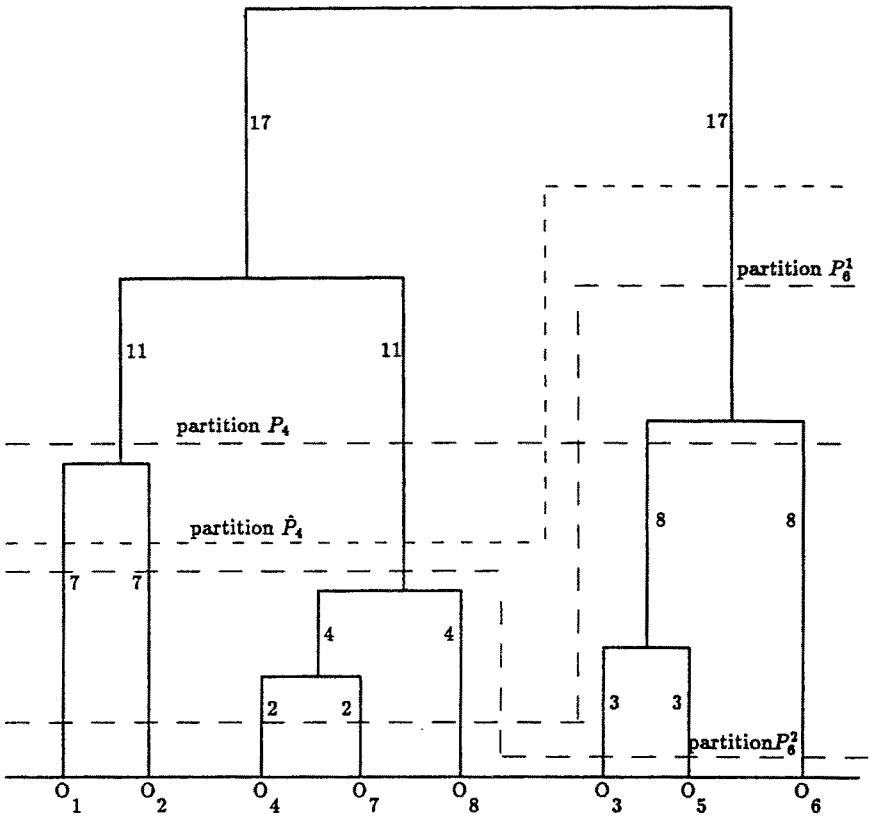

Figure 2. Dendrogram of Example 1.

4. A Property of Maximum Sum-of-splits Partitions

While maximum sum-of-splits partitions may differ from those of the single-linkage algorithm, there is a strong relationship between the clusters appearing in these partitions.

Theorem 2 *For any minimum spanning tree T and for every $M = 2,3,\ldots,N-1$, there is a maximum sum-of-splits partition P_M of O such that all its clusters belong to the set of $2N-1$ clusters of the single-linkage hierarchy obtained from T. Moreover, this is true for all maximum sum-of-splits partitions if there are no ties in the dissimilarity values.*

Proof. We first show that for a given minimum spanning tree T and for any given number of clusters M, there is a maximum sum-of-splits partition P_M such that $C_1, C_2, \ldots, C_M$ induce connected subgraphs of T. Consider a partition P_M such that this property does not hold. Each cluster C_i induces one or

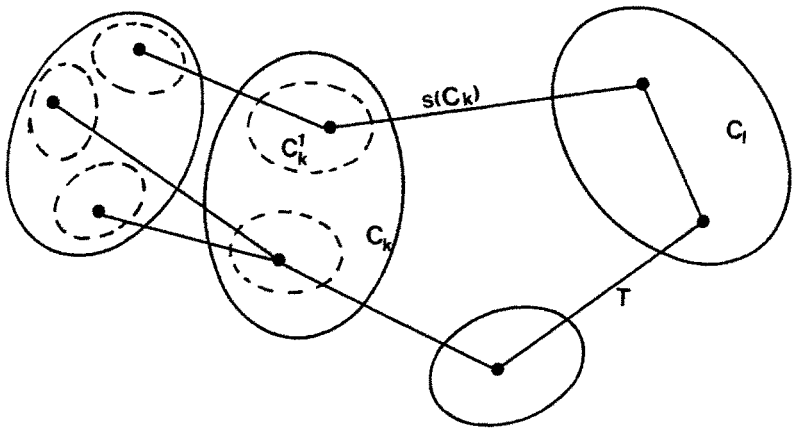

Figure 3. Connected subgraphs of T induced by partition P_M.

several connected subgraphs of T. Let $p > M$ denote the number of such subgraphs induced by all clusters of P_M. Let C_k be a cluster inducing several connected subgraphs and C_k^1 an induced connected subgraph such that $s(C_k^1) = s(C_k)$ (see Figure 3). Let C_l denote the cluster which includes the endpoint outside C_k of the edge defining $s(C_k)$. Note that $s(C_l) \leq s(C_k)$. Consider then the partition obtained from P_M by replacing C_k by $C_k \backslash C_k^1$ and C_l by $C_l \cup C_k^1$. Then $s(C_k \backslash C_k^1) \geq s(C_k)$ and $s(C_l \cup C_k^1) \geq s(C_l)$, hence the sum of the splits cannot decrease. If it increases, the partition P_M is not optimal. If it remains the same, we note that p has decreased by one unit and the result follows by induction.

We then show that the optimum value is independent of the minimum spanning tree which is chosen when it is not unique. Indeed, assume that T_1 and T_2 are minimum spanning trees. A maximum sum-of-splits partition P_M^1 for T_1 can be converted into a partition P_M^2 of not smaller sum-of-splits for T_2 using the above reasoning. As the converse is also true, the maximum sum-of-splits partitions P_M^1 and P_M^2 must be of equal value. Hence it suffices to reason on any minimum spanning tree T of G.

Next, fixing T, we show there is a maximum sum-of-splits partition $P_M = \{C_1, C_2, \ldots, C_M\}$ such that all its clusters belong to the single-linkage hierarchy H. A cluster C_k belongs to H if and only if its split $s(C_k)$ is such that $d_{rs} \leq s(C_k)$ for all edges $\{v_r, v_s\}$ of T such that v_r and v_s belong to C_k. We use induction on the number q of edges which do not satisfy this condition. Consider a partition P_M such that the condition is not satisfied, i.e., there is a cluster C_k such that $d_{rs} > s(C_k)$ for an edge $\{v_r, v_s\}$ of T such that v_r and v_s belong to C_k (see Figure 4). Let then T_k denote the subtree of T induced by the vertices of C_k and C_k^1 be a cluster associated with the vertices of a subtree of T_k obtained by deleting $\{v_r, v_s\}$ and such that $s(C_k^1) = s(C_k)$. Let C_l denote

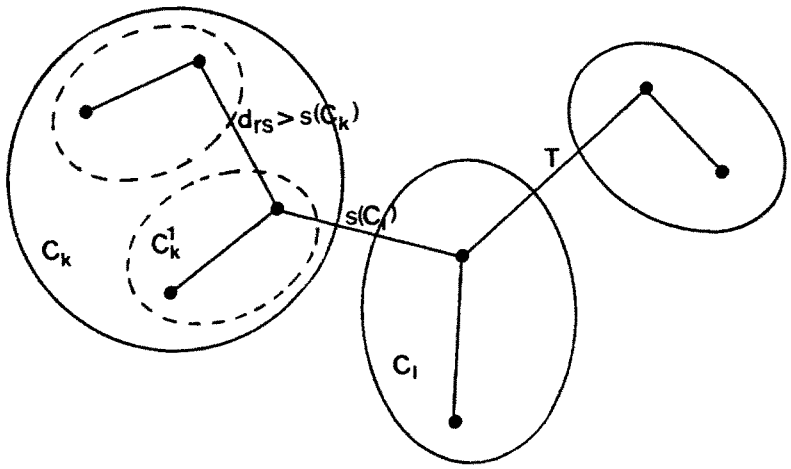

Figure 4. Clusters of H and of P_M.

again the cluster which includes the endpoint outside C_k of the edge defining $s(C_k)$. Consider then the partition obtained from P_M by replacing C_k by $C_k \backslash C_k^1$ and C_l by $C_l \cup C_k^1$. Again $s(C_k \backslash C_k^1) \geq s(C_k)$, $s(C_l \cup C_k^1) \geq s(C_l)$ and the sum of splits cannot decrease. If it increases, the partition $\{C_1, C_2, \ldots, C_M\}$ is not optimal. If it remains the same, we note that q has decreased by one and the result follows by induction.

Finally, assuming no ties in the dissimilarities, we recall that T is unique in that case (see e.g., Rosenstiehl 1967, Hubert 1974) and note that $s(C_k \backslash C_k^1) > s(C_k)$ or $s(C_l \cup C_k^1) > s(C_l)$ holds in points 1 and 3 of the proof. Hence all maximum sum-of-splits partitions must satisfy the stated condition. ∎

Let us note that the maximum sum-of-splits partition P_M need not be unique even if all dissimilarities are different in the chosen minimum spanning tree T. This is illustrated by example 1. The partitions $P_6^1 = \{ \{O_1\}, \{O_2\}, \{O_4\}, \{O_7\}, \{O_8\}, \{O_3, O_5, O_6\} \}$ and $P_6^2 = \{ \{O_1\}, \{O_2\}, \{O_3\}, \{O_5\}, \{O_6\}, \{O_4, O_7, O_8\} \}$ are both maximum sum-of-splits partitions with $ss(P_6^1) = ss(P_6^2) = 39$ (see again Figure 2).

The Dual Graph of a Dendrogram and an Algorithm

It is customary to represent the results of the single-linkage algorithm on a dendrogram: vertical lines correspond to entities and clusters, horizontal lines to fusions between them. The dendrogram can be associated with a planar graph $G_D = (V_D, E_D)$, partitioning the plane into faces. Notice that the

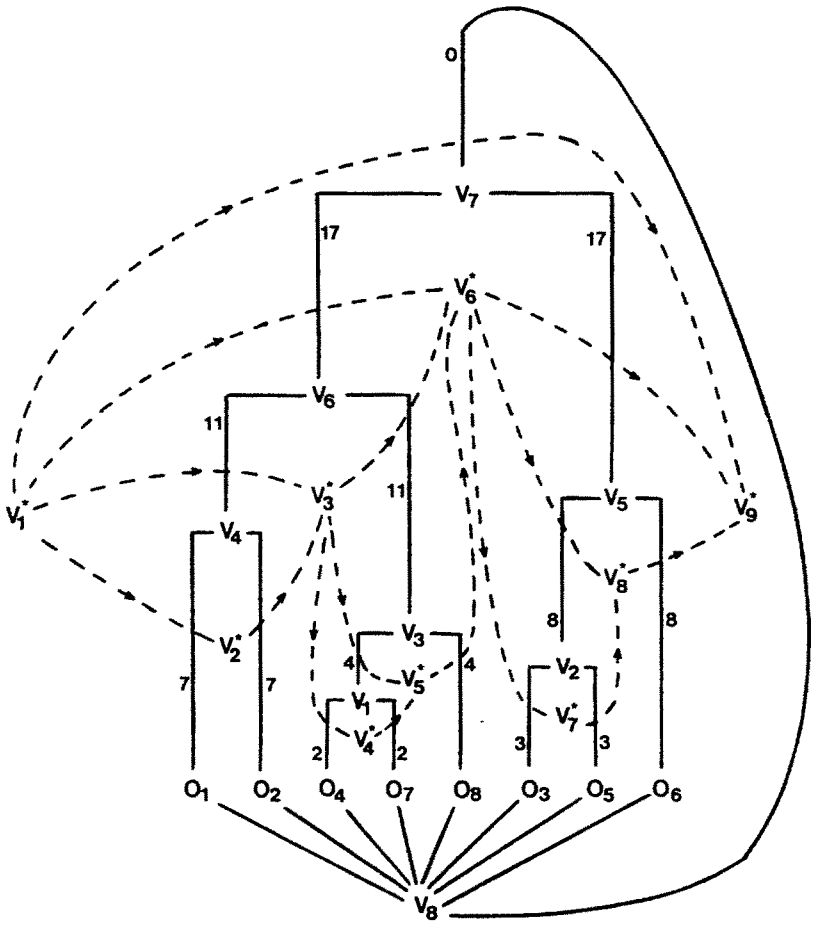

Figure 5. Dendrogram G_D and its dual G_D^*.

faces include the exterior infinite one. To this effect we modify the dendrogram by extending the vertical lines corresponding to entities down to a sink and adding a line on the right, outside the dendrogram, linking the sink and the middle of the highest horizontal line. Then intersections of more than 2 lines are vertices. Hence $V_D = \{v_1, v_2, \ldots, v_N\}$. Continuous lines between vertices are edges. See Figure 5 for a representation of graph G_D. This graph can easily be computed while building the dendrogram (with algorithm SLA) as follows:

(a) Associate a vertex v_k to each cluster C_{N+k} of the single-linkage hierarchy, $k = 1,2,\ldots,N-1$, and the vertex v_N to the sink.

(b) When fusing clusters C_i and C_j in cluster C_{N+k}, $k = 1,2,\ldots,N-1$, if C_i contains more than one entity then add the edge $\{v_{i-N}, v_k\}$ and add the edge $\{v_N, v_k\}$ otherwise; if C_j contains more than one entity then add the edge $\{v_{j-N}, v_k\}$ and add the edge $\{v_N, v_k\}$ otherwise.

(c) Add the edge $\{v_{N-1}, v_N\}$.

We can then consider the dual graph $G_D^* = (V_D^*, E_D^*)$ of G_D (see e.g., Bondy and Murty 1980) where a vertex v_j^* is associated with each face of G_D, and an edge joins v_j^* to v_k^* if and only if the corresponding faces are adjacent, i.e., have a common edge in their boundaries. The dual graph G_D^* has $N + 1$ vertices. Indeed, there is one face associated with each fusion and two exterior faces. It is known that the dual graph G^* of a graph G has the same number m of edges as G and can be built in $O(m)$ operations assuming the graph G is given by its adjacency lists (or, in other words, by the list of neighbors of each vertex) in clockwise order with respect to the planarity of G. In fact, it is enough to have all adjacency lists in clockwise order except one. Now, notice that all vertices of G_D have exactly three adjacent vertices except v_N, so that the clockwise order condition is easily satisfied for all vertices except v_N. So G_D^* can be computed in $O(N)$ operations since G_D has $2N - 1$ edges, each of them corresponding to a cluster of the single-linkage hierarchy. We assume that the vertex of the exterior face is located on the left of the dendrogram and has index 1, and that the vertex of the second rightmost face has index $N + 1$.

Moreover, edges of G_D^* can always be oriented from left to right, if the vertices v_j^* of the dual graph are located below the highest vertex of G_D on the boundary of their face, except for the first and the last ones v_1^* and v_{N+1}^*. From now on, G_D^* denotes this oriented graph, with arc-set U_D^*. Note that G_D^* is acyclic. We assume further that the vertices of G_D^* are labeled in topological order, i.e., in such a way that each arc (v_j^*, v_k^*) satisfies the condition $j < k$. Indeed, while building the dual graph and without modifying the complexity, one can always assign indices to the vertices as follows: if vertex v_l^* is located below vertex v_{N+k} of G_D, i.e., is associated with the fusion of cluster C_i (assumed to be the leftmost of the two in the dendrogram) and cluster C_j, l is equal to the order, in the dendrogram, of the leftmost entity of C_j. The dual G_D^* of G_D for example 1 is also represented on Figure 5.

The edges of the graph G_D are weighted by the splits of the clusters with which they are associated. These are equal to the heights of the horizontal lines corresponding to fusions of these clusters. Arcs (v_j^*, v_k^*) of the dual graph G_D^* are given the weight w_{jk} of the edges they are crossing. The resulting weighted graph for example 1 is reproduced in Figure 6.

Then, as proved in Theorem 3 below, the maximum sum-of-splits partitions P_M for $M = 2,3,\ldots,N-1$, correspond to the longest paths between the first and last vertices v_1^* and v_{N+1}^* of the dual graph G_D^* containing $2,3,\ldots,N-1$ arcs respectively.

These paths can be easily found using the following labeling algorithm. The labels λ_k^p for $k = 1,2,\ldots,N+1$, p integer and $p < k$ denote the value of

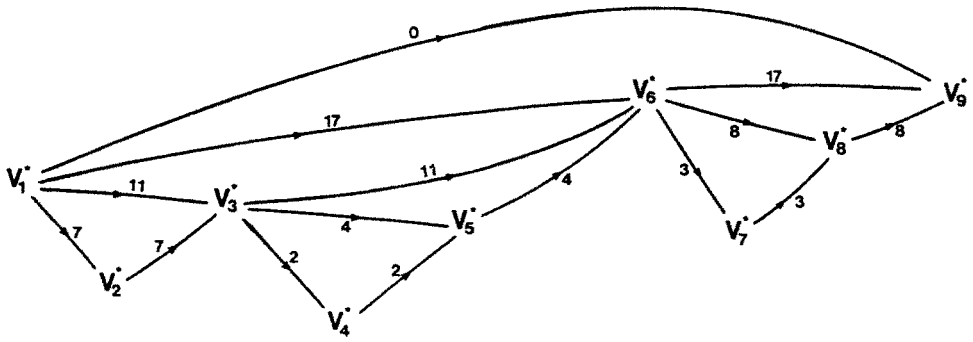

Figure 6. Weighted dual graph.

the longest path already found from v_1^* to v_k^* containing p arcs. These values are computed sequentially, as in the PERT method (see e.g., Lawler 1976) but taking also into account the numbers of arcs of the paths.

Algorithm DMSS (Dual Maximum Sum-of-Splits)

(1) Initialization. Set $\lambda_k^p = 0, k = 1,2,\ldots,N+1, p = 1,2,\ldots,N$.

(2) Current step.
For $k = 2,3,\ldots,N+1$ compute:
For each predecessor v_j^* of v_k^* do:
$\lambda_k^p = max\,(\lambda_k^p, \lambda_j^{p-1} + w_{jk})$ $p = 1,2,\ldots,j$,
Note by a pointer p_k^p an index j realizing this maximum.

(3) Maximum sum-of-splits partitions.
Recompose the longest paths $PA_M = (a_1,a_2,\ldots,a_M)$ for $M = 2,3,\ldots,N-1$ using the pointers recursively from v_{N+1}^* to v_1^*: set p to M, k to $N+1$ and l to p_k^p. Then until $p = 0$ repeat: $a_p \leftarrow (v_l^*, v_k^*)$; $p \leftarrow p-1$; $k \leftarrow l$; $l \leftarrow p_k^p$. Output the corresponding partitions using the dendrogram.

This algorithm can be accelerated, without modifying its worst-case complexity, by taking into account the lengths of the shortest paths from v_1^* to v_k^* for $k = 2,3,\ldots,N$. Details are given in Hansen, Jaumard and Musitu (1988).

We now show that algorithm DMSS solves the maximum sum-of-splits clustering problem.

Theorem 3 *Maximum sum-of-splits partitions P_M for $M = 2,3,\ldots,N-1$ can be computed in $\Theta(N^2)$ time.*

Proof. Each path of G_D^* from v_1^* to v_{N+1}^* corresponds to a partition of O into clusters of H. These clusters are those associated with the vertical lines of the

dendrogram crossed by that path. Conversely all partitions of O containing only clusters of H can be associated with such paths. Moreover, the sums of splits of these partitions are equal to the lengths of these paths. Then, from Theorem 2, any maximum sum-of-splits partition P_M can be associated with a path of G_D^* from v_1^* to v_{N+1}^*. So, the problem reduces to that of finding the longest paths between v_1^* and v_{N+1}^* containing $M = 2,3,\ldots,N-1$ arcs. This last problem is solved by algorithm DMSS. Correctness of this algorithm follows easily from the optimality principle of Dynamic Programming (see e.g., Bellman 1957, Bellman and Dreyfus 1962).

Regarding complexity, we first observe that building the graph G_D can be done while computing the single-linkage dendrogram without changing the complexity, i.e., in $\Theta(N^2)$, as described in the beginning of this section. Then, building the weighted dual graph can be easily done in $O(N)$.

We next show that algorithm DMSS is in $O(N^2)$. This is obvious for step 1. For step 2, we observe that it is easy to obtain a description of G_D^* by the list of the predecessors of its vertices in $O(N)$: the undirected dual graph G_D^* is given by its adjacency lists, so since the vertices of the (directed) graph G_D^* are labeled in topological order, the list of the predecessors of a vertex v_j^* ($j = 1,2,\ldots,N+1$) can be obtained from its adjacency list by selecting in it the vertices with index $k < j$. Then, as G_D^* has $2N-1 \equiv O(N)$ arcs and each of them is considered at most once in the computation of λ_k^p for each value of p, i.e., for $O(N)$ values, step 2 is also $O(N^2)$. In step 3 recomposition of each longest path takes M operations with $M = 2,3,\ldots,N-1$, i.e., $O(N^2)$ operations for all of them. Finally listing the entities of the clusters of the partitions $P_2,P_3,\ldots,P_{N-1}$ can be done in $O(N^2)$ if each vertex v_j of G_D has pointers to the leftmost and rightmost entities of the cluster associated with it and each entity has a pointer to the next entity on its right in the dendrogram. Hence, the complexity of algorithm DMSS, as well as that of the whole procedure, is in $O(N^2)$.

We finally note that the value of the sum of splits of a partition requires looking at the dissimilarities of all edges in the cocycles of its clusters, i.e., $O(N^2)$ operations. Therefore the proposed procedure has the best possible complexity up to a constant factor, i.e., it is in $\Theta(N^2)$. ∎

We now illustrate algorithm DMSS on example 1. The values of the parameters λ_k^p and of the pointers p_k^p are given in Table 2.

The longest paths of length $p = 1,2,\ldots,7$ are:

$p = 1$ (v_1^*, v_9^*) with value 0,
$p = 2$ (v_1^*, v_6^*, v_9^*) with value 34,
$p = 3$ $(v_1^*, v_3^*, v_6^*, v_9^*)$ with value 39,
$p = 4$ $(v_1^*, v_2^*, v_3^*, v_6^*, v_9^*)$ with value 42,
$p = 5$ $(v_1^*, v_2^*, v_3^*, v_6^*, v_8^*, v_9^*)$ with value 41,

Table 2. Illustration of Algorithm DMSS on Example 1

	λ_2^p	λ_3^p	λ_4^p	λ_5^p	λ_6^p	λ_7^p	λ_8^p	λ_9^p	λ_k^p/p
	7	11			17			0	1
		14	13	15	22	20	25	34	2
			16	18	25	25	30	39	3
p_2^p	1			18	22	28	33	42	4
p_3^p	1	2			22	25	31	41	5
p_4^p	—	3	3			25	30	39	6
p_5^p	—	3	3	4			28	38	7
p_6^p	1	3	3	5	5			36	8
p_7^p	—	6	6	6	6	6			
p_8^p	—	6	6	6	7	6	7		
p_9^p	1	6	6	6	8	8	8	8	
p_k^p/p	1	2	3	4	5	6	7	8	

$p = 6$ $(v_1^*, v_2^*, v_3^*, v_6^*, v_7^*, v_8^*, v_9^*)$ and
$(v_1^*, v_2^*, v_3^*, v_4^*, v_5^*, v_6^*, v_9^*)$ with value 39,
$p = 7$ $(v_1^*, v_2^*, v_3^*, v_4^*, v_5^*, v_6^*, v_8^*, v_9^*)$ with value 38.

The maximum sum-of-splits partitions for $M = 2,3,\ldots,N-1$ are:

$P_2 = \{\{O_1,O_2,O_4,O_7,O_8\},\{O_3,O_5,O_6\}\}$,
$P_3 = \{\{O_1,O_2\},\{O_4,O_7,O_8\},\{O_3,O_5,O_6\}\}$,
$P_4 = \{\{O_1\},\{O_2\},\{O_4,O_7,O_8\},\{O_3,O_5,O_6\}\}$,
$P_5 = \{\{O_1\},\{O_2\},\{O_4,O_7,O_8\},\{O_3,O_5\},\{O_6\}\}$,
$P_6^1 = \{\{O_1\},\{O_2\},\{O_4\},\{O_7\},\{O_8\},\{O_3,O_5,O_6\}\}$ and
$P_6^2 = \{\{O_1\},\{O_2\},\{O_3\},\{O_5\},\{O_6\},\{O_4,O_7,O_8\}\}$,
$P_7 = \{\{O_1\},\{O_2\},\{O_4\},\{O_6\},\{O_7\},\{O_8\},\{O_3,O_5\}\}$.

Note that: (i) three out of seven partitions (P_4, P_5 and P_7) are not identical to the partitions obtained by the single-linkage algorithm; (ii) partitions are not always hierarchical, see e.g., P_5, P_6^1 or P_6^2, P_7; (iii) optimal partitions are not necessarily unique even if all dissimilarity values are distinct.

6. Experimental Results

We now consider the analysis of a data set with the single-linkage algorithm (SLA) and the maximum sum-of-splits algorithm (DMSS) presented in

Section 5. Both algorithms have been implemented in Fortran 77 on a Sun 3/360S Microsystem. We use the SLINK algorithm of Sibson (1973) for SLA. Our program for the DMSS algorithm is described in Hansen, Jaumard and Musitu (1988) and is available upon request.

The data set concerns the principal nutrients in 27 foods (meat, fish and fowl). The matrix of dissimilarities can be computed (using Euclidean distance) from the data given in Hartigan (1975, p. 87). Some of the optimal partitions ($\hat{P}_8$, $\hat{P}_{13}$, $\hat{P}_{19}$) which differ from the ones obtained with the single-linkage algorithm are indicated on the dendrogram of Figure 7 (small-dashed lines). For a given value M, partitions obtained by SLA may have different sum-of-splits values due to identical dissimilarity values. Two of these partitions (P^1_{13}, P^2_{13}) are also represented on Figure 7 (large-dashed lines). The sum-of-splits values of the partitions are given between parentheses on the right side of the figure.

These results suggest the following conclusions, corroborated by the study of a dozen more data sets:

(a) Partitions obtained by SLA and DMSS tend to have the same sum-of-splits values for small M.
(b) Differences in the values of the sum-of-splits increase with the heterogeneity of the dissimilarity values.

7. Conclusions

The single-linkage algorithm is among the most used in cluster analysis and has many desirable properties (see e.g., Jardine and Sibson 1971). It can be viewed as a method to find maximum split partitions for any number M of clusters. We show in this paper that the single-linkage algorithm also maximizes the sum of the splits of all $2N - 1$ clusters of the hierarchy that it defines, but does not maximize the sum of the splits of all clusters in a partition. We therefore study this last problem and provide a $\Theta(N^2)$ algorithm to determine maximum sum-of-splits partitions into M clusters for $M = 2,3,\ldots,N-1$. Results with several data sets from the literature show that better maximum sum-of-splits partitions than those obtained by the single-linkage algorithm are usually obtained for some values of M, but quite often the partitions of the single-linkage algorithm are also optimal for the sum-of-splits criterion. Therefore the single-linkage algorithm appears to be a good heuristic to solve the maximum sum-of-splits clustering problem. However, the single-linkage algorithm is also in $\Theta(N^2)$. So both the maximum split and the maximum sum-of-splits clustering problems can be solved exactly at small cost. This is done by using first the single-linkage algorithm and then by applying the algorithm of this paper to the dual graph of the single-linkage dendrogram.

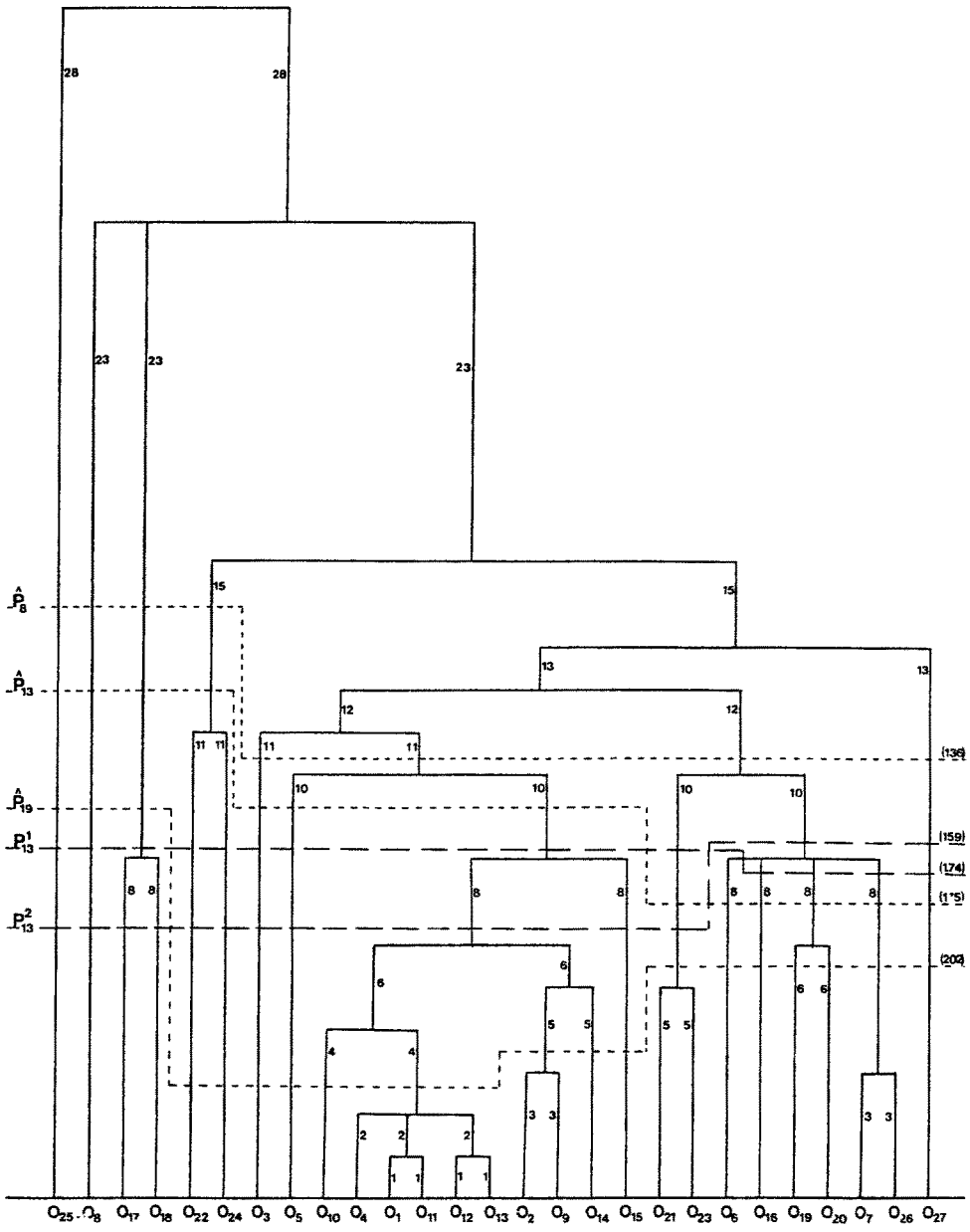

Figure 7. Single-linkage dendrogram and sum-of-splits partitions of a data set.

References

BELLMAN, R. E. (1957), *Dynamic Programming*, Princeton: Princeton University Press.

BELLMAN, R. E., and DREYFUS, S. E. (1962), *Applied Dynamic Programming*, Princeton: Princeton University Press.

BONDY, J. A., and MURTY, U. S. R. (1980), *Graph Theory with Applications*, New York: North Holland.

BRUCKER, P. (1978), "On the Complexity of Clustering Problems," in *Optimization and Operations Research*, Lecture Notes in Economics and Mathematical Systems, 157, eds. M. Beckman and H. P. Kunzi, Heidelberg: Springer, 45-54.

DELATTRE, M., and HANSEN, P. (1980), "Bicriterion Cluster Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 2(4)*, 227-291.

GORDON, A. D. (1981), *Classification: Methods for the Exploratory Analysis of Multivariate Data*, New York: Chapman and Hall.

HANSEN, P., and DELATTRE, M. (1978), "Complete-link Cluster Analysis by Graph Coloring," *Journal of the American Statistical Association, 73*, 397-403.

HANSEN, P., and JAUMARD, B. (1987), "Minimum Sum of Diameters Clustering," *Journal of Classification, 4*, 215-226.

HANSEN, P., JAUMARD, B., and MUSITU, K. (1988), "Algorithm DMSS," in preparation.

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

HUBERT, L. (1974), "Spanning Trees and Aspects of Clustering," *British Journal of Mathematical and Statistical Psychology, 27*, 14-28.

HUBERT, L. (1977), "Data Analysis Implications of Some Concepts Related to the Cuts of a Graph," *Journal of Mathematical Psychology, 15*, 199-208.

JARDINE, N., and SIBSON, R. (1971), *Mathematical Taxonomy*, New York: Wiley.

LAWLER, L. (1976), *Combinatorial Optimization: Networks and Matroids*, New York: Holt, Rinehart and Winston.

LECLERC, B. (1977), "An Application of Combinatorial Theory to Hierarchical Classification," in *Recent Developments in Statistics*, eds. J. R. Barra et al., North Holland, 783-786.

MONMA, C., and SURI, S. (1989), "Partitioning Points and Graphs to Minimize the Maximum or the Sum of Diameters," in *Proceedings of the Sixth International Conference on the Theory and Applications of Graphs*, eds. Y. Alavi, G. Chartrand, O. R. Oellermann and A. J. Schwenk, New York: Wiley.

ROSENSTIEHL, P. (1967), "L'arbre minimum d'un graphe," in *Théorie des Graphes*, Rome, I.C.C., ed. P. Rosenstiehl, Paris: Dunod, 357-368.

SIBSON, R. (1973), "SLINK: An Optimally Efficient Algorithm for the Single-link Cluster Method," *The Computer Journal, 16*, 30-34.

ZAHN, C. T. (1971), "Graph-theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Transactions on Computers, C-20*, 68-86.