CHAPTER 3 - SYNOPTIC CLUSTERING

3.1 INTRODUCTION

The problem of synoptic clustering is very different from the search for "natural" clusters or patterns. In the latter problem, the input data must be faithfully respected; in the synoptic clustering problem convenient output is of greater interest.

This Chapter will chiefly focus on Ward's method, or the minimum variance method. For most applications this method can be recommended for the summarization of data. Section 3.2 will attempt to justify this statement : it will informally describe the minimum variance agglomerative method, and will detail properties of this method which are of practical importance.

Mathematical properties of the minimum variance method are detailed in section 3.4. Preceding this, section 3.3 establishes the equivalence of agglomerative algorithms based on distances (or dissimilarities) and algorithms based on cluster centres. The former is the traditional approach, while the latter is used in the recently-proposed, fast algorithms described in later sections. Apart from the minimum variance method, other agglomerative methods which can be implemented with very little alteration are concurrently described. The geometric methods all have the property of defining a cluster centre, or cluster representative.

An essential property of the fast algorithms - a precondition for their use, which depends on the cluster criterion - is discussed in section 3.5. Sections 3.6 and 3.7 describe the fast algorithms, and prove the major performance results.

3.2 MINIMUM VARIANCE METHOD IN PERSPECTIVE

Agglomerative clustering methods have been motivated by graph theory - leading to linkage-based methods - or by geometry. This generalization, which is true for the more commonly used methods, indicates the most productive frameworks for studying clustering methods. In geometric methods, the cluster centre may be used for subsequent agglomerations. Alternatively, inter-cluster dissimilarities may be used throughout, and therefore these methods may be implemented using the Lance-Williams dissimilarity update formula (see Table 1, section 3.3 below) as in the case of linkage-based methods.

In order to specify an agglomerative criterion simultaneously in terms of cluster mean vectors, and in terms of dissimilarity, it is necessary to adopt a particular dissimilarity. In this Chapter, it will be assumed that the Euclidean distance is employed. Restricting the choice of dissimilarity to this distance is rarely inconvenient in practice. The Euclidean distance is often the most natural choice of distance, and a Euclidean space offers a well-known and powerful standpoint for analysis.

The variance or spread of a set of points (i.e. the sum of squared distances from the centre) has been the point of departure for specifying many clustering algorithms. Many of these algorithms, - iterative, optimization algorithms as well as the hierarchical, agglomerative algorithm - are briefly described and appraised in Wishart (1969). The use of variance in a clustering criterion links the resulting clustering to other data-analytic techniques which involve a decomposition of variance. Principal components analysis, for example, seeks the principal directions of elongation of the multidimensional points, i.e. the axes on which the projections of the points have maximal variance. Using a clustering of the points with minimal variance within clusters as the cluster-criterion is, perhaps, the most suitable criterion for two concurrent but compli-

mentary analyses of the same set of points. The reality of clusters of projected points resulting from the principal components analysis may be assessed using the cluster analysis results; and the interpretation of the axes of the former technique may be used to facilitate interpretation of the clustering results.

The search for clusters of maximum homogeneity leads to the minimum variance criterion. Since no coordinate axis is privileged by the Euclidean distance, the resulting clusters will be approximately hyperspherical. Such ball-shaped clusters will therefore be very unsuitable for examining straggly patterns of points. However, in the absence of information about such patterns in the data, homogeneous clusters will provide the most useful condensation of the data.

The following properties make the minimum variance agglomerative strategy particularly suitable for synoptic clustering.

(1) As discussed in section 3.4, the two properties of cluster homogeneity and cluster separability are incorporated in the cluster criterion. For summarizing data, it is unlikely that more suitable criteria could be devised.

(2) As in the case of other geometric strategies, the minimum variance method defines a cluster centre of gravity. This mean set of cluster members' coordinate values is the most useful summary of the cluster. It may also be used for the fast selection and retrieval of data, by matching on these cluster representative vectors rather than on each individual object vector.

(3) A top-down hierarchy traversal algorithm may also be implemented for information retrieval. Using a query vector, the left or right subtree is selected at each node for continuation of the traversal (-it is best to ensure that each node has precisely two successor nodes in the construction of the hierarchy).

Such an algorithm will work best if all top-down traversals through the hierarchy are of approximately equal length. This will be the case if and only if the hierarchy is as "symmetric" or "balanced" as possible (see Fig. 3.1). Such a balanced hierarchy is usually of greatest interest for interpretative purposes also : a partition, derived from a hierarchy, and consisting of a large number of small classes, and one or a few large classes, is less likely to be of practical use.

For such reasons a "symmetric" hierarchy is desirable. It has been shown, using a number of different measures of hierarchic symmetry, that the minimum variance (closely followed by the complete link) methods generally give the most symmetric hierarchies (see Murtagh, 1984). This is an important, albeit "post festum ", property of hierarchies produced by the minimum variance method.

(4) Unlike other geometric agglomerative methods - in particular the centroid and the median methods (see definitions, Table 1, below) - the sequence of agglomerations in the minimum variance method is guaranteed not to allow inversions in the cluster criterion value. Inversions or reversals (Fig. 3.2) are inconvenient, can make interpretation of the hierarchy very difficult, and will be investigated in the following section.

(5) Finally, computational performance has until recently favoured linkage-based agglomerative criteria, and in particular the single linkage method. The computational advances described below for the minimum variance method (and other methods) make it increasingly attractive for practical applications involving large amounts of data. Some of the algorithms to be discussed lend themselves to a parallel implementation : as parallel machine architectures become more widely available, research in this direction should be particularly fruitful.

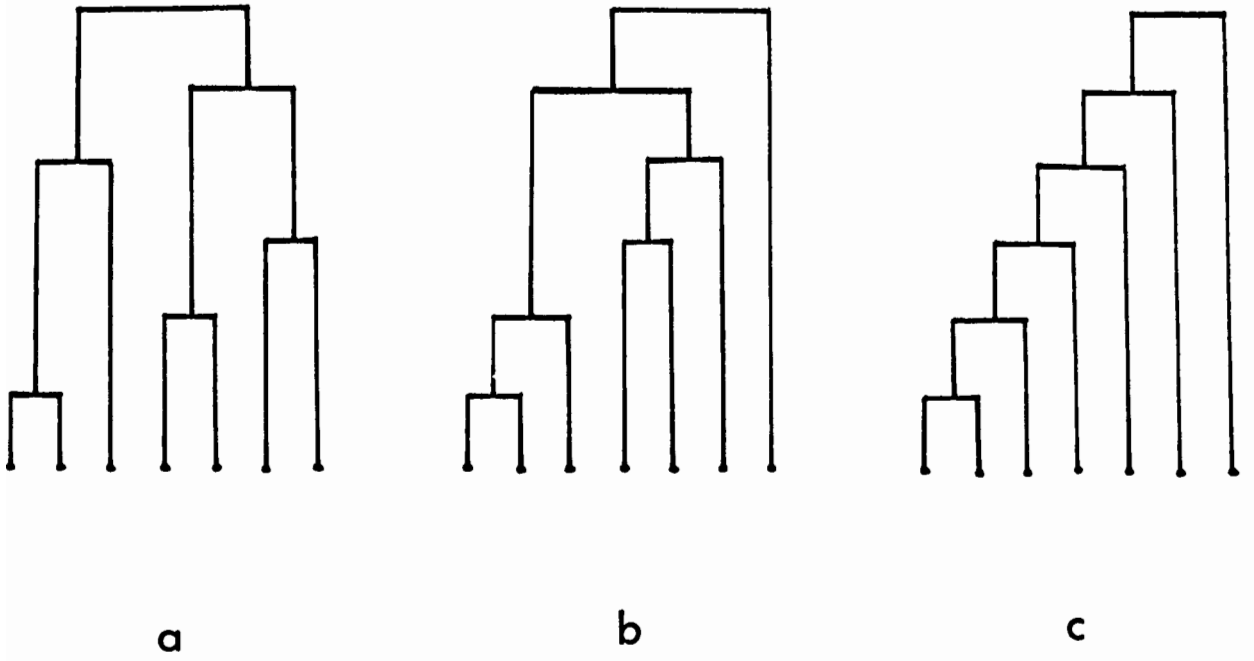

Fig. 3.1 - Three binary hierarchies with n=7 terminal or leaf nodes; (a) is symmetric, (c) is as asymmetric as possible and (b) presents an intermediate case.

In this Chapter, references to many of the original proposals for the algorithms described may be consulted in Murtagh (1983).

Fig. 3.2 - Alternative representations of a hierarchy with an inversion (reversal) in cluster criterion values.

3.3 GEOMETRIC AGGLOMERATIVE METHODS

Hierarchic agglomerative algorithms may be conveniently broken down into linkage (or graph theoretic) methods - the single, complete, weighted and unweighted average linkage methods - and cluster centre (or geometric) methods - the centroid, median, and minimum variance methods. The latter may be specified either in terms of dissimilarities, alone, or alternatively in terms of cluster centre coordinates and distances (dissimilarities, in the case of the minimum variance method). Let us prove this in the case of the median method.

Let a and b be two points (i.e. m-dimensional vectors: objects or cluster centres) which have been agglomerated, and let c be another point. From the Lance-Williams dissimilarity update formula, using squared Euclidean distances, we have :

$$\begin{aligned} d^2(a\cup b,c) &= d^2(a,c)/2 + d^2(b,c)/2 - d^2(a,b)/4 \\ &= (a-c)^2 + (b-c)^2/2 - (a-b)^2/4 \end{aligned} \qquad (1)$$

The new cluster centre is (a+b)/2, so that its distance to point c is

$$(c- (a+b)/2)^2 \qquad (2)$$

That these two expressions are identical is readily verified. The correspondence between these two perspectives on the one agglomerative criterion is similarly proved for the centroid and minimum variance methods.

For these geometric methods, and with suitable alterations for graph methods, the following algorithm is an alternative to the general dissimilarity-based algorithm described in Chapter 1 (which may be described as a "stored dissimilarities approach").

Hierarchical clustering methods (and aliases).	Lance and Williams dissimilarity update formula.	Coordinates of centre of cluster, which agglomerates clusters i and j.	Dissimilarity between cluster centres g_i and g_j.
Single link (nearest neighbour).	a(i)=0.5 b=0 c=-0.5 (More simply: min{d(i,k),d(j,k)}.)	-	-
Complete link (diameter).	a(i)=0.5 b=0 c=0.5 (More simply: max{d(i,k),d(j,k)}.)	-	-
Group average (average link, UPGMA).	a(i)=|i|/(|i|+|j|) b=0 c=0	-	-
McQuitty's method (WPGMA).	a(i)=0.5 b=0 c=0	-	-
Median (Gower's method, WPGMC).	a(i)=0.5 b=-0.25 c=0	$g=(g_i+g_j)/2$	$\|\|g_i-g_j\|\|^2$
Centroid (UPGMC).	a(i)=|i|/(|i|+|j|) $b=-\|i\|.\|j\|/(\|i\|+\|j\|)^2$ c=0	$g=(\|i\|g_i+\|j\|g_j)/(\|i\|+\|j\|)$	$\|\|g_i-g_j\|\|^2$
Ward's method (minimum variance, error sum of squares).	a(i)=(|i|+|k|)/(|i|+|j|+|k|) b=-|k|/(|i|+|j|+|k|) c=0	$g=(\|i\|g_i+\|j\|g_j)/(\|i\|+\|j\|)$	$(\|i\|.\|j\|/(\|i\|+\|j\|)).\|\|g_i-g_j\|\|^2$

Notes : |i| = number of objects in cluster i.

g_i is a vector in m-space (where m is set of variables); either initial point or cluster centre.

||.|| is the norm in the Euclidean metric.

Lance and Williams recurrence formula ([ ]: absolute difference):

$d(i \cup j,k)=a(i).d(i,k)+a(j).d(j,k)+b.d(i,j)+c.[d(i,k)-d(j,k)]$.

Algorithm A. Stored Data Approach

Step 1 : Examine all interpoint dissimilarities, and form cluster from two closest points.

Step 2 : Replace two points clustered by representative point (centre of gravity) or by cluster fragment.

Step 3 : Return to Step 1, treating clusters as well as remaining objects, until all objects are in one cluster.

In Steps 1 and 2, "point" refers either to objects or clusters, both of which are defined as vectors in the case of geometric methods. This algorithm is justified by storage considerations, since we have O(n) storage required for n initial objects and O(n) storage for the n-1 (at most) clusters. In the case of graph methods, the term "fragment" in Step 2 refers to a connected component in the case of the single link method and to a clique or complete subgraph in the case of the complete link method. The overall complexity of the above algorithm is at best $O(n^3)$: the repeated calculation of dissimilarities in Step 1, coupled with O(n) iterations through Steps 1, 2 and 3. Note however that this does not take into consideration the extra processing required in a graph method, where "closest" in Step 1 is defined with respect to graph fragments.

Apart from lessened storage requirements, Algorithm A can also be justified in that it can be made computationally very efficient by searching for the clusters, in Step 1, in a restricted, local region only. This will be looked at below.

3.4 MINIMUM VARIANCE METHOD : MATHEMATICAL PROPERTIES

The minimum variance method produces clusters which satisfy compactness and isolation criteria. These criteria are incorporated into the dissimilarity, noted in Table 1, as will now be shown.

In Ward's method, we seek to agglomerate two clusters, c_1 and c_2, into cluster c such that the within-class variance of the partition thereby obtained is minimum. Alternatively, the between-class variance of the partition obtained is to be maximized (see Lemma 1, below). Let P and P* be the partitions prior to, and subsequent to, the agglomeration; let p_1, p_2, ... be classes of the partitions:

$$P = \{ p_1, p_2, \ldots, p_k, c_1, c_2 \}$$

$$P^* = \{ p_1, p_2, \ldots, p_k, c \} .$$

Finally, let i denote any individual or object, and I the set of such objects. In the following, classes (i.e. p or c) and individuals (i.e. i) will be considered as vectors or as sets: the context will be sufficient to make clear which is the case. Total variance of the cloud of objects in m-dimensional space is decomposed into the sum of within-class variance and between-class variance. This is Huygen's theorem in classical mechanics (for proof, see Lemma 1 below), and for the two partitions we have respectively:

$$\mathrm{Var}(I) = \mathrm{Var}(P) + \sum_{p \in P} \mathrm{Var}(p)$$

$$\mathrm{Var}(I) = \mathrm{Var}(P^*) + \sum_{p \in P^*} \mathrm{Var}(p)$$

Hence :

$$\mathrm{Var}(P) + \mathrm{Var}(p_1) + \ldots + \mathrm{Var}(p_k) + \mathrm{Var}(c_1) + \mathrm{Var}(c_2)$$
$$= \mathrm{Var}(P^*) + \mathrm{Var}(p_1) + \ldots + \mathrm{Var}(p_k) + \mathrm{Var}(c) \quad .$$

Therefore :

$$\mathrm{Var}(P^*) = \mathrm{Var}(P) + \mathrm{Var}(c_1) + \mathrm{Var}(c_2) - \mathrm{Var}(c) \quad .$$

In agglomerating two classes of P, the variance of the resulting partition (i.e. Var(P*)) will necessarily decrease: therefore in seeking to minimize this decrease, we simultaneously achieve a partition with maximum between-class variance. The criterion to be optimized is then :

$$\mathrm{Var}(P) - \mathrm{Var}(P^*)$$
$$= \mathrm{Var}(c) - \mathrm{Var}(c_1) - \mathrm{Var}(c_2)$$
$$= \frac{|c_1| \, . \, |c_2|}{|c_1| + |c_2|} \; ||c_1 - c_2||^2 \qquad \text{(see Lemma 2 below),}$$

which is the dissimilarity given in Table 1. This is a dissimilarity which may be determined for any pair of classes of partition P; and the agglomerands are those classes, c_1 and c_2, for which it is minimum. Having shown what we set out to prove, we now turn attention to two results which were used in the foregoing.

Lemma 1 : T = W + B, where T is the total variance of the set of points, I; W is the within-class variance of a partition of I; and B is the between-class variance of this partition.

Proof : Let partition P be defined as $\{p_1, p_2, \ldots, p_k\}$.
We are to prove that :

$$\mathrm{Var}(I) = \mathrm{Var}(P) + \sum_{p \in P} \mathrm{Var}(p)$$

i.e.

$$\frac{1}{n}\sum_{i\in I}(i-g)^2 = \sum_{p\in P}\frac{|p|}{n}(p-g)^2 + \frac{1}{n}\sum_{p\in P}\sum_{i\in p}(i-p)^2$$

where g is the grand mean of the n objects : $g = \frac{1}{n}\sum_{i\in I} i$; and $|p|$ is the cardinality of class p. Note that p is used interchangeably to denote centre of gravity (a vector), and the set whose centre of gravity this is.

Rewriting the rightmost term in the above expression gives us :

$$\sum_{p\in P} Var(p) = \frac{1}{n}\sum_{p\in P}\sum_{i\in p}(i-p)^2 = \sum_{i\in I}\frac{1}{n}((i-g)-(p-g))^2$$

$$= \sum_{i\in I}\frac{1}{n}(i-g)^2 + \sum_{i\in I}\frac{1}{n}(p-g)^2 - 2\sum_{i\in I}\frac{1}{n}(i-g)(p-g)$$

$$= \frac{1}{n}\sum_{i\in I}(i-g)^2 + \sum_{p\in P}\frac{|p|}{n}(p-g)^2 - \frac{2}{n}\sum_{p\in P}|p|(p-g)^2$$

$$= \frac{1}{n}\sum_{i\in I}(i-g)^2 - \sum_{p\in P}\frac{|p|}{n}(p-g)^2$$

$$= Var(I) - Var(P) .$$

Corollary: For any choice of P, T is constant; therefore any choice of P which maximizes W (a measure of class compactness) simultaneously minimizes B (a measure of class isolation).

Lemma 2: $Var(c) - Var(c_1) - Var(c_2) = \frac{|c_1|.|c_2|}{|c_1|+|c_2|}\ ||c_1 - c_2||^2$

where $c = c_1 \cup c_2$.

Proof: Consider the variance of class c which is decomposed into classes c_1 and c_2. By Lemma 1 we have:

$$\mathrm{Var}(c) = \mathrm{Var}(\{c_1,c_2\}) + \mathrm{Var}(c_1) + \mathrm{Var}(c_2).$$

Hence, we are to show that

$$\mathrm{Var}(\{c_1,c_2\}) = \frac{|c_1|.|c_2|}{|c_1|+|c_2|} \; ||c_1 - c_2||^2$$

Since c is the centre of gravity of vectors c_1 and c_2 we have :

$$\mathrm{Var}(\{c_1,c_2\}) = |c_1| \; . \; ||c_1-c||^2 \; + \; |c_2|.||c_2-c||^2$$

Writing

$$c = \frac{|c_1|.c_1 + |c_2|.c_2}{|c_1| + |c_2|}$$

and substituting gives the desired result.

Corollary: If c_1 and c_2 are singleton classes, then $\mathrm{Var}(\{c_1,c_2\})$ $= \frac{1}{2} ||c_1 - c_2||^2$ (i.e. the Euclidean distance between a pair of classes is twice their variance).

3.5 REDUCIBILITY PROPERTY

The reducibility property lays down the condition under which the fast algorithms, reviewed in subsequent sections, may be constructed locally, by carrying out agglomerations in restricted regions of the space of objects, but such that the hierarchy arrived at is nonetheless exact. This reducibility property is also closely related to the problem of reversals or inversions in hierarchic, agglomerative algorithms, i.e. $d(a \cup b, c) \not\geqslant d(a, b)$ for some clusters or objects a, b, and c and where a and b agglomerate to constitute $a \cup b$ (see Fig. 3.2). Following the statement of the reducibility property, three other equivalent or derived expressions of this property will be examined. The importance of this property for clustering algorithms will then be studied.

Reversals and the reducibility property.

Consider the agglomeration of clusters or objects, a and b, into $c = a \cup b$. Consider also some other cluster or object c'. The reducibility property is then:

$$d(a,b) \leqslant \inf\{d(a,c'), d(b,c')\}$$
$$\Rightarrow \inf\{d(a,c'), d(b,c')\} \leqslant d(a \cup b, c'). \qquad (3)$$

Verbally, the agglomeration of a and b cannot produce a cluster $c = a \cup b$ which will be closer to cluster c' than was either a or b. If such were the case, then it is possible that c and c' might subsequently agglomerate at a smaller proximity than in the case of the agglomeration of a and b: i.e. there would be an inversion or reversal. Thus, if a clustering strategy satisfies the reducibility property, inversions cannot arise. This is proved, for any given clustering strategy, by using the Lance-Williams dissimilarity update formula. Before looking at Ward's method as an example, it will be convenient to derive an equivalent ex-

pression of the reducibility property.

If $d(a,b) \leqslant \rho \leqslant \inf \{ d(a,c'), d(b,c') \}$, then from (3) we have: $\rho \leqslant \inf \{ d(a,c'), d(b,c') \} \leqslant d(c,c')$. Therefore we can write:

$$\left.\begin{array}{l} d(a,b) \leqslant \rho \\ d(a,c') \geqslant \rho \\ d(b,c') \geqslant \rho \end{array}\right\} \Rightarrow d(c,c') \geqslant \rho \qquad (4)$$

This form of the reducibility property allows easy verification. Taking Ward's method, we have:

$$d(c,c') = \left((|a|+|c'|)d(a,c') + (|b|+|c'|)d(b,c') - |c'|\, d(a,b) \right) / \left(|c|+|c'| \right)$$

Using (4), the right hand side is

$$\geqslant \rho \cdot \left((|a|+|b| + |c'| \right) / \left(|c|+|c'| \right)$$

which is necessarily $\geqslant \rho$. Therefore $d(c,c')$ is also $\geqslant \rho$. The proof that the single, complete, and average linkage methods satisfy the reducibility property is similarly proved. In the case of the centroid and median methods, however, it is not possible to guarantee that $d(c,c') \geqslant \rho$.

Two further expressions for the reducibility property will be derived. In (3) consider some ρ greater than or equal to the terms on both sides of the implication. We have:

$$\begin{array}{l} d(a,b) \leqslant \rho \\ d(c,c') \leqslant \rho \end{array}$$

and we must necessarily have:

$$d(a,c') \leqslant \rho \quad \text{or} \quad d(b,c') \leqslant \rho \tag{5}$$

Since the contrary does not necessarily hold, if we are using a geometric method where each of a, b, and c are reduced to a point, we have that

$$B_\rho(c) \subseteq B_\rho(a) \cup B_\rho(b) \tag{6}$$

where $B_\rho(\ )$ is the hypersphere of specified centre and of radius ρ. More generally, B_ρ may be defined as the set of all possible points within a dissimilarity ρ of the cluster members. Because of the latter expression, the reducibility property may also be referred to as a space contraction property. Note that since the reducibility property is satisfied by the complete link method, the property of spatial contraction is used here in a different sense to its use by Lance and Williams (1967).

Reciprocal nearest neighbours and the reducibility property.

Let the nearest neighbour graph (NN-graph) be defined as a set of points, p, whose directed edges {(p,NN(p))} are such that NN(p) is the nearest neighbour of p. For any point p, three cases can be distinguished: see Fig. 3.3. In case III, where q = NN(p), and p = NN(q), points p and q are referred to as mutual or reciprocal nearest neighbours (RNNs).

The NN-graph can be defined at all stages of the construction of the hierarchy for a geometric clustering method, - where the vertices consist of remaining, unclustered, initial points and cluster representative points - and with suitable additional operations can also be used for a graph clustering method. Before looking at algorithms for these different areas, consider first the initial situation where the set of n multi-

dimensional points are given. If the reducibility porperty is verified by a clustering method, the merging of RNNs i and j into cluster $i \cup j$ requires the updating of the NN-graph only for those points which had i or j as NN (cf. Case II, Fig. 3.3). More importantly it means that all RNNs i and j can be simultaneously merged, without effecting the RNN properties of other parts of the NN-graph.

These two corollaries of the reducibility property will be used in the algorithms to be studied in the next 2 sections.

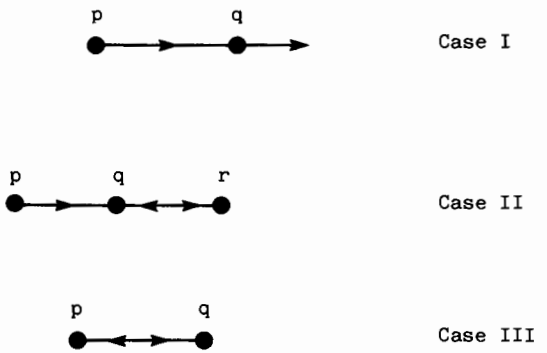

Fig. 3.3 - Three situations of interest in the NN-graph.

3.6 MULTIPLE CLUSTER ALGORITHM

The following algorithm (Algorithm B) provides $O(n^2)$ worst case time and $O(n)$ space results for geometric strategies which have the additional property (as will be seen following Algorithm B) of defining inter-cluster dissimilarity as distance. From Table 1 (3rd column), this is the case for the centroid and median methods but not for Ward's method. These complexity results therefore will apply to the approximate centroid and median hierarchies obtained using Algorithm B. (The approximation relates to the non-verification of the reducibility property by these methods. For exact results, RNNs would need to be considered in decreasing order of closeness, in order to allow for the effects produced by an agglomeration on other RNNs. Instead, for computational efficiency, the RNNs will be processed here in the order that they are found. Discrepancy with exact algorithms for the centroid and median methods will be obtained when there is a reversal or inversion, and will rarely be inconvenient from a practical point of view.)

Algorithm B. Parallel clustering

Step 1. (Re)determine all NNs and RNNs.

Step 2. Agglomerate all RNNs, replacing with cluster point.

Step 3. Go to Step 1 until one point remains.

Let t be the number of iterations carried out ($t \leqslant n-1$). On the first execution of Step 2, if the number of RNNs found is r_1, then r_1 agglomerations are carried out, and $n-r_1$ points remain. We have: $1 \leqslant r_1 \leqslant \lfloor n/2 \rfloor$, since at least one point, and at most half the point-set, will be RNNs. Since there are precisely $n-1$ agglomerations, if we find on successive iterations $r_2, r_3, \ldots, r_t$ RNNs in Step 2, we have:

$$r_1 + r_2 + \ldots\ldots + r_t = n-1 \qquad (7)$$

Good time complexity results for this algorithm depend (as will be seen) on whether or not the number of NN calculations to be carried out following any agglomeration is constant.

Complexity: $O(n^2)$ computation and $O(n)$ storage for approximate median and centroid methods.

Proof: In Step 1, n dissimilarity calculations are carried out, each of which requires n-1 calculations. The RNNs are determined in $O(n)$ operations, by a straightforward scan of the n NNs. In Step 2, r_1 RNNs are found. Let the number of points p corresponding to Case II of Fig. 3.3 be upper-bounded by a constant for each such pair of RNNs: this supposition is discussed below. Then there will be less than or equal to $r_1 + c.r_1$ new NN calculations to be performed (i.e the NNs of the r_1 new cluster centres will be required, as will the new NNs of the - at most- c/2 points which correspond to point p of Case II in Fig. 3.3). Similarly, on the following iteration, there will be at most $r_2 + c.r_2$ new NNs to be determined. Thus, in all, the number of NN calculations is bounded from above by

$$
\begin{aligned}
& n + r_1(c+1) + r_2(c+1) + \ldots + r_t(c+1) \\
&= n + (c+1)(n-1) && \text{using (7)} \\
&= O(n)
\end{aligned}
$$

Each such calculation will take less than or equal to n-1 dissimilarity calculations, giving overall complexity of $O(n^2)$.

Proof of supposition: The number of points which can have a given point as NN in the Euclidean plane is less than or equal to 6 (consider the limit case where the given point is placed at the centre of a circle of radius ρ, on whose perimeter the 6 points are placed such that all ad-

jacent pairs are precisely a distance ρ apart). In higher dimensional spaces, use may be made of the fact that the Euclidean distance is upper-bounded by the Chebyshev (∞ or maximum coordinate) distance. In the L_∞ metric, hyperspheres of radius ρ become (from an L_2 perspective) cubes of edge length 2ρ. The number of points which can simultaneously have a given point as NN in m-dimensional Chebyshev space is then 3^m-1 (i.e. the number of cubes which are adjacent to a given cube). For given m, this is constant. Although we may be reasonably happy that this lower bound holds also for the Euclidean distance, it is a great deal more difficult to prove this. Day and Edelsbrunner (1984) may be consulted for this proof.

Unfortunately the above result does not hold for Ward's method, where a dissimilarity between cluster centres results from each agglomeration (cf. column 3, Table 1). A worst case of O(n) NN calculations must be assumed to follow each agglomeration, which leads to overall $O(n^3)$ worst case time.

Due to the fact that each agglomeration leaves certain points unmatched (cf. Case II, Fig. 3.3), Algorithm B has been dubbed "algorithme des célibataires" in the Vol. VII, No. 2, 1982 issue of the journal Les Cahiers de l'Analyse des Données which focussed on these new approaches to clustering using NNs, RNNs, and NN-chains. Because the number of célibataires after each batch of agglomerations increases exponentially with the dimensionality of the point-space, it appears that Algorithm B is only feasible for relatively low dimensional spaces (about 4 or 5 at most). Algorithm C, to be discussed in the next section, is more practicable. Algorithm B, however, allows agglomerations to be carried out in as parallel a fashion as possible and thus may well be a natural choice of algorithm as parallel processors become more widely available.

An exact version of this algorithm for the centroid and median methods may be obtained with a small amount of extra processing. The smallest distance is found at each agglomeration. This will require only O(n) extra processing, given the list of NNs, and the smallest dissimilarity is necessarily a RNN pair. The agglomeration is carried out, and the list of NNs updated in O(n) time. There are n-1 iterations, and hence the overall time complexity remains $O(n^2)$. Algorithm B differs in that the RNN pair of least dissimilarity is not obtained (thus saving O(n) calculations on each iteration); and that all updates of the list of NNs can be carried out on the reduced set of clusters/objects which follows each batch of agglomerations (thus saving further time).

3.7 SINGLE CLUSTER ALGORITHM

Rather than carrying out as many agglomerations as possible at each iteration, the algorithms to be discussed in this section only carry out one agglomeration per iteration.

The single cluster algorithm is based on the NN-chain. Starting with an arbitrary object or cluster, i, this is defined as

$$i, NN(i)=j, NN(j)=k, \ldots, NN(p)=q, NN(q)=p.$$

Let it be assumed that no two dissimilarities are equal (arbitrary resolving of such cases will be discussed below). Then the following three propositions hold for NN-chains:

Proposition 1 : Inter-object/cluster dissimilarities monotonically decrease as we proceed along the NN-chain.

Proposition 2 : The final link always connects a RNN pair.

Proposition 3 : The NN-chain cannot contain a circuit of more than two nodes.

Proof 1 : If we have $NN(i)=j$ and $NN(j)=k$, for $i \neq k$, then necessarily: $d(i,j) > d(j,k)$, since otherwise i would be the NN of j, contrary to construction.

Proof 2 : For some p, in determining $q=NN(p)$, we must have either $q \notin$ NN-chain, in which case a new link is grown onto the NN-chain; or $q \in$ NN-chain; by Proposition 1, this can only be the object/cluster which

precedes p in the NN-chain.

Proof 3 : If we had a cycle, for some i, j, ... , p, q:

i, NN(i)=j, ... , NN(p)=q, NN(q)=i

then Proposition 1 would be violated.

Owing to Proposition 1, we may say that the NN-chain is grown towards increasing density, since inter-point dissimilarity - hence sparseness - at the start of the NN-chain is greater than that at the end.

In practice, some dissimilarities might be equal. In any implementation of the algorithm to be described below, arbitrary resolution of such cases must be provided for. In particular, a circuit such as

i, NN(i)=j, NN(j)=k, NN(k)=i

where d(i,j) = d(j,k) = d(k,i)

must be prevented in the NN-chain.

Algorithm C, as follows, is suitable for any geometric strategy.

Algorithm C. NN-chain clustering

Step 1. Select a point arbitrarily.

Step 2. Grow the NN-chain from this point until a pair of RNNs are obtained.

Step 3. Agglomerate these points, replacing with a cluster point.

Step 4. From the point which preceded the RNNs, or from an arbitrary point if there is no such point, go to Step 2 until only one point remains.

If i is the first point selected, we obtain the sequence

i, NN(i)=j, NN(j)=k, ... , NN(o)=p, NN(p)=q, NN(q)=p

(Note that i and j could constitute a RNN pair). Points p and q are merged, and a new point replaces them. This contraction of the NN-chain is followed by a further set of growths starting from point o (or from an arbitrary point if the RNN pair were the only two points in the NN-chain). Algorithm C is exact if agglomeration of a RNN pair doesn't affect the RNN properties of any other objects and clusters, i.e. if the reducibility property is satisfied by the agglomerative strategy used.

Complexity: Algorithm C is optimal for all geometric methods; i.e. it requires $O(n^2)$ computation and $O(n)$ storage.

Proof: Let a growth of the NN-chain refer to the adding of a link, and a contraction refer to the agglomeration of a pair of RNNs. Algorithm C is seen to be a series of intermixed growths and contractions. The number of contractions is n-1 (i.e. the number of agglomerations). The number of growths cannot exceed 3n-3: i.e. the number of nodes incorporated into the NN-chain can never exceed the n initial points, plus the n-1 cluster points created, which gives a total of 2n-2 links; and a final set of n-1 links must be considered which allow a RNN pair to be made out of the final link in the NN-chain. Now, each growth requires 1 NN calculation. Each contraction requires a constant number of operations. Therefore, the overall complexity - assuming $O(n)$ effort for a NN calculation - is $O(n^2)$. Storage of the NN-chain, the original data, and the cluster points, is altogether $O(n)$.

In growing a link onto a NN-chain, when n_1 points are in the NN-chain and

n_2 points are not ($n_1+n_2 \leqslant n$), the number of NN calculations to be carried out is n_2+1: dissimilarities between the last point in the NN-chain and the n_2 points not in the chain must be determined, as also the dissimilarity between the last and the second last points in the NN-chain. It is fruitless to examine dissimilarities with any of the other n_1 points since these must be greater that the NN dissimilarity required.

For graph (or linkage) methods, where inter-cluster dissimilarity cannot be calculated in O(1) time unless we have the entire set of dissimilarities directly accessable, Algorithm C may be amended as follows.

Algorithm D. NN-chain algorithm using stored dissimilarities.

Step 1. Select an object arbitrarily.

Step 2. Determine and store all inter-object dissimilarities.

Step 3. Grow the NN-chain from the object chosen, until a pair of RNNs are obtained.

Step 4. Agglomerate these objects.

Step 5. Update the dissimilarity table, using the Lance-Williams formula.

Step 6. From the node in the NN-chain which preceded the RNNs, or from an arbitrary node (object or cluster) if the NN-chain is empty, go to Step 3 until only one node remains.

Clearly, $O(n^2)$ storage is required here. As before, there are O(n) growths of the NN-chain, each requiring O(n) updating of the dissimilarity table. In total, therefore, computational complexity is $O(n^2)$.

Algorithm D provides time-optimal algorithms for the weighted and unweighted average linkage methods (UPGMA, WPGMA). It does so also for any general strategy based on the Lance-Williams recurrence formula, and will be exact if the agglomerative strategy satisfies the reducibility property.

It does not appear that the $O(n^2)$ computational complexity of Algorithm D can be further improved. However, the complexity of Algorithm B is seen to be $O(nf)$ where $O(f)$ is the complexity of finding a nearest neighbour of a point and a brute-force approach to this subproblem is $O(n)$. Efficient linear and sub-linear algorithms can instead be used to obtain nearest neighbours. Such algorithms have been reviewed in Chapter 2.

3.8 REFERENCES

W.H.E. DAY and H. EDELSBRUNNER, Efficient algorithms for agglomerative hierarchical methods. Journal of classification 1, 7-24, 1984.

G.N. LANCE and W.T. WILLIAMS, A general theory of classificatory sorting strategies. I. Hierarchical systems. The Computer Journal 9, 373-380, 1967.

F. MURTAGH, A survey of recent advances in hierarchical clustering algorithms. The Computer Journal 26, 354-359, 1983.

F. MURTAGH, Structures of hierarchic clusterings: implications for information retrieval and for multivariate data analysis. Information Processing and Management (in press, 1984).