A Survey of Algorithms for Contiguity-constrained Clustering and Related Problems

F. MURTAGH

*Department of Computer Science, University College Dublin, Dublin 4, Ireland**

A large number of non-parametric clustering algorithms from a wide range of applications in the social sciences, earth sciences, pattern recognition, and image processing, are critically appraised. These algorithms all have the common property of seeking to use a relational – usually contiguity – constraint, in addition to proximity information. The constraint is necessary in many applications for the visualisation of clustering results. The primary objective of this survey is to sketch out the major algorithmic paradigms in current use, with a view towards facilitating the task of algorithm design in this area.

1. INTRODUCTION

One major theme in clustering research over the past two decades has been the automatic classification of quantitatively described objects, without any constraint as to which pairs of such objects might ultimately find themselves in the same class. A second major trend in clustering work has been where there is such an inherent or an imposed representational constraint. This second area of clustering arises naturally in the analysis of point patterns, and lately it has become of increasing interest in the analysis of data in the geo-sciences. Even though the objectives of contiguity-constrained clustering algorithms may differ in pattern recognition, image processing, urban and regional studies, psychometrics, and so on, underlying principles are often shared, and valuable lessons may be learnt from other disciplines for the design of new algorithms. In this article we review general-purpose algorithms from these different areas. All of the algorithms have the function of segmenting (or regionalising or zoning) a set of objects, each described by a descriptor vector; or alternatively the algorithms use contiguity information, and so can be easily adapted for the foregoing problem. The algorithms are discussed under three major headings but it is not to be thought that they are only applicable in the application-areas chiefly described (social sciences and earth sciences in Section 2; pattern recognition and image processing in Section 3; again, social sciences in Section 4). On the contrary, the aim of this paper is to collect together and contrast algorithms, any number of which might be suitable for a given application.

Contiguity-constrained clustering uses proximities between objects, defined in descriptor space, and also takes into account contiguous neighbourhoods. Depending on the application, the contiguous neighbourhood is defined in different ways (references to the various definitions to be described can be found in the context of the algorithms discussed below). In image processing, where the image consists of pixels characterised by grey-level intensity values, the eight neighbouring pixels (east, north-east, north, etc.) are suitable candidates. Similarly with agricultural data, the terrain which is characterised by crop yields or chemical constituents may be subdivided into square parcels and the neighbourhood of a parcel may be defined as its eight adjacent parcels. With point patterns, a radius may be used to define the neighbourhood of point $i: N(i) = \{j | d_{ij} \leq r\}$, and j is said to be contiguous to i. In order to remove the influence of scale, a neighbourhood may alternatively be defined as the k nearest neighbours of an object. In general, when the objects do not comprise the squares of a regular grid, it is convenient to express the contiguity relationship as a binary matrix, with a contiguity value $c_{ij} \in \{0, 1\}$ defined on all pairs of objects. Such a matrix can be externally defined by the user, for example, in the case of contiguities between bordering countries (characterized, perhaps, by socio-economic attributes) or other basic spatial units.

One final approach to defining contiguity is described in Section 2; this is where a continuous contiguity measure is used, rather than the discrete 0–1 alternative. In this case, the contiguity matrix may be defined as the dissimilarites between objects in the representational space (usually the Euclidean plane).

In Section 2 general contiguity-constrained algorithms will be described. It will be assumed that the contiguity relation is given here by a binary matrix, unless otherwise mentioned. The principal algorithms discussed in this section include the single and complete linkage hierarchical clustering methods and Ward's minimum variance method.

In Section 3, even though analysis of point patterns raises very different problems from the multidimensional clustering problem, its implicit use of contiguity information (especially in its use of neighbourhoods) makes it valuable for suggesting possible algorithms. As will be seen, many of the algorithms in this area are related to the single-linkage hierarchical clustering method. More precisely, they construct subsets of a minimal spanning tree, the definition of which is first generalized for asymmetric dissimilarities. The possibility of using algorithms motivated by those used for point pattern analysis in other areas is further discussed in the Conclusion.

Although the algorithms discussed for the analysis of interaction data in Section 4 appear in practice not to require an explicit constraint, the problems for which they are designed share many aspects of the contiguity problem. In particular they require geographic representation. One feature of this area is that centroid-based

* Current address: European Space Agency, Space Telescope European Coordinating Facility, ESO, Karl-Schwarzschild-Str. 2, 8046 Garching, W. Germany.

have been widely used, and this section therefore provides an interesting case-study where the hierarchic centroid-based algorithms of Section 2, and the single linkage-based procedures of Section 3, have both been profitably applied to the same problem.

2. CLUSTERING SUBJECT TO CARTOGRAPHIC REPRESENTABILITY

One approach to the problem of contiguity-constrained clustering is to take traditional algorithms which have proved their worth in other areas and to enforce geographic or planar representability by incorporating contiguity. We will consider suggested hierarchic (and, briefly, non-hierarchic) algorithms which work on both objects' coordinates in descriptor space and their contiguity relation in (planar) representation space. Specialised optimisation algorithms can, of course, be designed, but contiguity-constrained algorithms which are applicable to a wide range of types of data and application areas will be chiefly focused on. Constraints other than contiguity (or a similar relation) have also been suggested. A constraint on the maximum number of members in a cluster, for example, which is mentioned below, may however be replaced by the *a posteriori* choice of suitable clusters from a hierarchy. The problem of a linear constraint on objects to be clustered (e.g. events in time, or depth of geological samples) has also been directly addressed by some authors,[1-3] and a solution to this problem can be sought using the more generally applicable techniques described in this and the following sections.

Hierarchic, agglomerative algorithms construct a series of partitions of the object set. With each partition, some (possibly suboptimal) value of a compactness or connectivity criterion is associated. If the stepwise agglomerations are constrained to be between contiguous clusters, the problem of inversions (reversals or non-monotonic increase/decrease in cluster criterion value) is likely. This is when $d(p \cup q, r) \ngtr d(p, q)$, for three clusters p, q and r, where p and q agglomerate to form $p \cup q$, and where the measure of compactness is related to the dissimilarity, d, between clusters. Inversions tend to be prominent in contiguity-constrained agglomerative methods since a previously forbidden merger between two very similar classes may be permitted by changes in the contiguity relation. The presence of inversions in a hierarchy is disadvantageous: it makes difficult the interpretation of partitions, and the definition of dissimilarity between classes (see Fig. 1). Only two of the traditional hierarchic clustering methods (many of which may be specified in terms of the Lance–Williams dissimilarity update formula;[4] see Jambu and Lebeaux[5] for an extended formula) can be amended to permit agglomerations between contiguous clusters, and simultaneously guarantee that no inversions will arise. These methods are the single and complete linkage methods.

The contiguity-constrained single linkage method[6] is as follows: at each agglomeration, fuse together the two clusters of least interconnecting dissimilarity, such that this dissimilarity is between a pair of contiguous objects. Initially all clusters are singletons. Each agglomeration in this method is necessarily between a pair of contiguous objects. Therefore, given the contiguity graph where each

Fig. 1. Alternative representations of hierarchy with an inversion (reversal) in cluster criterion values.

edge connecting a pair of contiguous objects is weighted by the dissimilarity (in descriptor space) between the objects, it is seen that the minimal spanning tree of the weighted contiguity graph may be obtained and subsequently transformed into the single-linkage hierarchy. A simple proof that the contiguity-constrained single-linkage hierarchy cannot present inversions is to replace the dissimilarities between all pairs of non-contiguous objects by some arbitrarily large value. The construction of the single-linkage hierarchy on this amended set of dissimilarities is well defined (in the sense that at all stages the traditional algorithm can be employed and, assuming the contiguity graph is connected, infinite dissimilarities will never be used as cluster criterion – connectivity – values). As in the case of the usual single-linkage method, it has been found that this method has a pronounced tendency to 'chain', i.e. to successively agglomerate singletons to one, large cluster in each partition.[7,8] Efficient algorithms for constructing a constrained single-linkage hierarchy are available in the $O(n^2)$ algorithm of Murtagh,[9] where n is the number of objects; or if the number of contiguous edges, m, is significantly less than n^2, in the algorithms of Cheriton and Tarjan[10] or Yao[11] for minimal spanning trees. The latter algorithms require $O(m \log \log n)$ worst case time, and may be followed by the conversion of the minimal spanning tree into the single-linkage hierarchy in $O(n)$ time.[12] For the clustering of geographic units, which implies that the graph will be planar, the time taken to construct the minimal spanning tree can be further improved to $O(n)$ worst case time.[10]

An alternative approach for contiguity-based agglomerative clustering allows agglomeration of any pair of clusters such that there exists a contiguity link between at least one member of each of the clusters. Of the major hierarchical methods, only the complete link method excludes the possibility of inversions when constrained in this manner. This is proved by Ferligoj and Batagelj,[13] and an example of application of the constrained complete linkage method to geographical regionalisation is to be found in Fischer.[8]

For practical application, both the single and the complete linkage methods suffer disadvantages, the former being over-lax in its criterion for creating classes and the latter being over-demanding. Ward's minimum-variance method has often been favoured as a generally applicable agglomerative strategy. In the area of contiguity-constrained clustering, a constrained version of this method has frequently been proposed in the past decade. Webster and Burrough[14] use Ward's method in the context of the same definition of contiguity between classes as was used in the inversion-free constrained complete-linkage method (i.e. dynamic or weak contiguity, as opposed to the strong definition of inter-class contiguity used in the constrained single-linkage method). The constrained minimum-variance method of the last reference was applied to soil parcels. Openshaw[15] similarly uses a number of definitions of compactness, related to Ward's minimum variance. A constrained Ward's method, with minor variations, is also used in Refs. 7, 16–18. Lebart's algorithm (and Roche's application to urban segmentation using sociological data) employ also a cluster size restraint which, once achieved, causes the cluster to be entirely removed from all further stages of the clustering. None of these algorithms precludes the possibility of inversions. Ferligoj and Batagelj[13] explicitly remove this possibility by proposing two suitable sets of coefficients in the Lance–Williams dissimilarity update formula such that an update formula resembling that of Ward's method is arrived at.

All of the proposals for a constrained Ward's method (with the exception of the last-mentioned reference) suffer from the difficulty inherent in interpreting a hierarchy with inversions. Notwithstanding the greater ease of cartographic representation, the problem of validating the results obtained remains difficult. While the compactness of clusters can be given a precise definition (variance of a cluster, average within-cluster distances, etc.), attempting to simultaneously optimize compactness and contiguity essentially requires user-specification of the relative importance of these two objectives. This is especially evident when another approach to incorporating contiguity information is used: two dissimilarities are employed – the dissimilarity defined in the descriptor space, and the geographical (or other relational) distance. Webster and Burrough[14] use various additive and multiplicative combinations of these two dissimilarities. Perruchet[19, 20] and Murtagh[21] use products of the two, but with separate recalculation of dissimilarities for each newly formed class. In all of these approaches, different relative weightings of the dissimilarities used will, evidently, entirely alter the resulting classification. Despite these inherent difficulties – relating to interpretability, and to the need for user-intervention in controlling the clustering – the algorithms discussed here worked adequately for particular data sets. Further empirical work is called for in order to ascertain which of these algorithms are to be most recommended for general-purpose applications.

Finally, mention will be made of non-hierarchic algorithms. Such methods are based on a preset number of classes; an arbitrarily defined initial partition (if no other choice is available); and iterative relocation (or exchange) of objects between clusters, while the partition criterion is improved. Note however that the group to which the object first belonged must not be disconnected; and that the group which the object is a candidate for joining must be contiguous to the object. Algorithms of this type are discussed in Refs. 8, 13 and 22.

3. MODE SEEKING

Hierarchical clustering algorithms and iterative, relocatory algorithms of the k-means type are generally based on inter-object dissimilarities. In point pattern analysis inter-point dissimilarities have also been used. Zahn's[23] approach to distinguishing subgroups of point patterns used the minimal spanning tree, and a number of post-processing techniques on the minimal spanning tree allowed analysis of a comprehensive range of types of pattern. Two extensions of this approach have motivated a good deal of work: on the one hand, the minimal spanning tree approach has been simplified by the use of a short(est) spanning path approach which achieves many of the same results with greater computational ease;[24–26] and on the other hand, other graph-theoretic structures have been used – in low dimensional spaces only – to which the minimal spanning tree is often related as a special case.[27–30] One difficulty with distance-based procedures is that any regularity in the data may give rise to many identical distances and to subsequent degenerate or misleading cluster results.

In order to circumvent this difficulty, an alternative approach has been pursued, especially for the analysis of point patterns. This makes use of valuations on the points under examination, with in some cases the additional use of interpoint dissimilarities. Therefore the algorithms may be said to be based either on node valuations alone, or on both node valuations and edge valuations. We will begin with algorithms in the first category.

Node weights used in pattern recognition, usually related to a supposed probability density function of point occurrence, have generally involved an estimate of density at each point. Among such node weights are the following.

(1) $|N(i)|$ where $N(i) = \{j | d_{ij} \leqslant r\}$.

$N(i)$ is the neighbourhood of point i, defined here as the set of points within radius r of i. The weight of node i is the cardinality of its neighbourhood (Refs. 31–33).

(2) $1/k \quad \Sigma \{d_{ij} | j \in N(i)\}$

where $N(i)$ is the set of k nearest neighbours of i. The weight of node i, here, is the average distance to the k-nearest neighbours; it is a measure of 'potential', i.e. the inverse of density (Ref. 34; see also 35).

(3) $\Sigma \{ \exp(-d_{ij}^2/w^2) | j = 1, \ldots, n; j \neq \mathrm{i}\}$

where w is some constant. This weight of node i is defined with reference to all other points, distant points contributing very little (Ref. 36; a similar idea is used in Ref. 37).

(4) In a problem not related to point pattern recognition, nodes corresponding to geographical regions have associated weights defined by per capita income (an analogue of point density where income replaces number of points).[38]

(5) In image processing, nodes corresponding to pixels may be weighted by the grey-level intensity at that point; or by a measure of edge gradient at that pixel (the latter may be useful for contour extraction).[39]

The use of dissimilarity d in the above is almost invariably Euclidean, the most natural choice for visual patterns of points. Let f_i be the weight associated with node i, using any of the above definitions.

The most straightforward approach to the clustering of node-valued graphs is to use a single threshold: nodes of density weight greater (or potential weight less) than the threshold are members of the same cluster, if they are in addition contiguous to at least one other member of the cluster. By decreasing the threshold in the case of densities, or by increasing it in the case of potential, a hierarchy of embedded classes is obtained which may be represented by a generalised skyline plot similar to that used for dendrograms (Katz and Rohlf[36] use such a diagram).

The clustering brought about by thresholding can also be expressed in terms of more traditional distance-based clustering . Define $\delta_{ij} = \infty$ if i and j are not contiguous; otherwise define $\delta_{ij} = -\min\{f_i, f_j\}$ where f is a density. As values of f are examined in increasing order of magnitude, i will be connected to j only if both f_i and f_j are greater than the density threshold (not to be confused with the contiguity threshold of definition (1) above, which is set prior to the search for clusters). This clustering method may be viewed as a constrained single-link method. Clustering by thresholding in the manner described is a common technique in image processing (see in particular the related work of Haralick and Dinstein);[40] it has been used for histogram segmentation (further discussed below);[41] and it has also been used for wealth data for geographical regions.[38]

A particular implementation, and alternative output of the clustering, is employed by Kittler:[42,43] starting with an arbitrary node, this node is connected to a contiguous neighbour such that -min $\{f_i, f_j\}$, $j \in N(i)$, is maximised; continuing from this neighbour node, all nodes in the given set are successively processed. The succession of values of the criterion function gives a useful idea of the presence of modes and valleys in the point density.

The use of node weights (such as point densities, attributes of populations or states, etc.) is an intuitively clear starting point from which to carry out the automatic grouping process. But the use of inter-point distances, while being fraught with difficulty when many distances are identical, none the less allows a more fine-tuned analysis: for example, in the threshold-based clustering described above, no account is taken as to whether a new addition to a cluster is closely related to one or to many of the cluster members. In order to allow for varying degrees of relationship, a dissimilarity may be re-created from the node weights. One possibility for this is to construct directed arcs defined by $\delta_{ij} = f_j - f_i$ where i and j are contiguous. Therefore if $f_j > f_i$ then δ_{ij} is directed from i to j, while if $f_j < f_i$ then the arc is negatively weighted, and so is directed from j to i. Rather than the difference in densities, as this dissimilarity coefficient is, the density gradient (difference in density per unit distance) has usually been peferred.[32,33,39] This is given by $\delta_{ij} = (f_i - f_j)/d_{ij}$, where d is the Euclidean distance and δ is again an asymmetric dissimilarity. A generalization of the single-linkage method (or the minimal spanning tree) has been used for such dissimilarities. It is to connect i to j if δ_{ij} is positive and maximum among nodes j which are contiguous to i, i.e. to construct components such that the density gradient is always upwards. It is easily verified that each such component is a directed tree, so long as no δ_{ij} equals zero. In order to facilitate subsequent labelling and other processing of the components, cycles in the directed graph must be prevented, and arbitrarily directed edges are formed for $\delta_{ij} = 0$ following a test that a cycle will not result. Note that in this approach each component nominates a unique 'centre' or local peak in density. It has also been proposed that local valleys in density are equally revealing of structure in the data.[33] Such an alternative viewpoint of the data may be carried out by simply defining δ_{ij} as the negative of the mode-oriented approach.

A similar approach – determining components which are directed trees – has been used in image processing. Narendra and Goldberg[39] define as a weight at each pixel (node) a measure of edge gradient (the *edge value* is the difference in intensities between contiguous pixels; and *edge gradient* is the maximum such value between a pixel and its neighbours). Having thus a value for f_i, the asymmetric dissimilarity δ_{ij} is constructed and the directed tree formed in the manner described above. Another very different application of this directed forest approach has also been successfully employed, as follows.[44–46] A histogram of intensities often permits visually different parts of the image to be distinguished – different modes in the histogram correspond to distinct, but significantly numerous, sets of pixel intensities. The gradient climbing procedure, used in point pattern recognition, also allows the modes of the histogram to be determined. Smoothing of the histogram might be required – using for instance a 3-point moving average – and Wharton[46] suggests an 'adaptive smoothing' where non-mode parts of the histogram (below a user-specified threshold value) alone are smoothed in this way. For 4-band Landsat data, a 4-dimensional generalization of this approach has been employed by constructing a 4-dimensional histogram. This is simply a grid of regular cells in 4-dimensional space, each containing the frequency of occurrence of associated 4-valued pixel intensity vectors.

The dissimilarity constructed in the foregoing examples has been anti-symmetric: $\delta_{ij} = -\delta_{ji}$. A different asymmetric dissimilarity will be employed in the analysis of flow data (Section 4). An asymmetric coefficient may also be constructed for point pattern recognition. Such a dissimilarity was used by Ozawa:[47] $\delta_{ij} = f_i \exp(-bd_{ij})$ where b is some scale constant. This dissimilarity will yield different values for δ_{ij} and for δ_{ji} depending on density defined at i and at j.

4. GEOGRAPHICAL INTERACTION DATA

Transaction flow matrices are square, asymmetric matrices which arise in many of the social sciences. Examples of the flows or interactions involved in such tables are industrial inputs and outputs, journal citations, internal migrations, occupational mobility and trip distributions.[48] In the last three of these areas of application, we might wish to relate the clusters found to the geographic locations of the basic spatial units which the flows refer to. Therefore contiguous clusters of basic spatial units are often required. Two types of clustering problem may be considered. Consider the case of journey-to-work data, with a given set of zones and associated numbers of cross-boundary journeys. We may wish to ascertain nodal or 'central' zones (i.e. those that receive large numbers of workers), or alternatively to carry out a regionalisation of the given zones into a smaller set of homogeneous areas. For these two different problems, two approaches have been suggested. A variant of the single-linkage method has been proposed for the former problem – the determining of nodal zones. Faithful representation of the asymmetric character of the interaction matrix is the primary objective, and one disadvantage of this approach is the 'chaining' side-effect of single-linkage clustering. For the second problem – creating homogeneous zones – variants of the average linkage method have been used. A disadvantage here is the conflict between the clusters of zones and the often asymmetric characteristics of these zones (i.e. inflows greater than outflows or vice versa).

In both cases a standardisation of the given flows is carried out, in order to adjust for disproportionate flow in large zones. In the case of the compact clustering, this has been achieved by dividing every element of the flow array by the corresponding row and column sums. In the case of the directed single-linkage procedure, this standardisation is iterated – row and column sums are recalculated until they are equal.

The directed linkage procedure involves one of three generalizations of the single-linkage method for dealing with asymmetric proximities (here: the standardized flows) discussed by Hubert.[49] The strong components of the directed graph are the sets of mutually reachable nodes (or zones): each node can be reached from another in the same component if there is a series of consistently directed arcs from one to the other. As in the case of the single-linkage method, a dendrogram may be constructed, corresponding to the components formed at differing thresholds of proximity. This approach to the clustering of flow data has been used in Refs. 48, 50–52. An efficient algorithm for obtaining the strong components at any given level has been discussed by Tarjan:[53] for a directed graph of m arcs and n nodes, this algorithm allows the hierarchy to be obtained in $O(m \log n)$ time.

For constructing a hierarchy of compact clusters, an algorithm proposed by Domengès is as follows.[54] A symmetric matrix is constructed by summing the (i, j)th and (j, i)th elements, for all i and j. Next, the symmetric matrix is standardised in the manner described above by dividing the (i, j)th element by the product of the associated row and column sums. Finally the sequence of agglomerations takes place by successively seeking the greatest standardised symmetric flow between regions. Let the symmetric matrix be defined from the given flow matrix by $s_{ij} = f_{ij} + f_{ji}$. When an agglomeration takes place, the (unstandardised) flows to and from the new region, c, equal the sum of flows to and from the sub-clusters a and b: $s_{cc'} = s_{ac'} + s_{bc'}$, for any other region, c'. If s_c and $s_{c'}$ are the totals of rows c and c' (or columns: the matrix has been made symmetric before all agglomerations), then the agglomerations take place on standardised values, s^*:

$$\begin{aligned} s^*_{cc'} &= s_{cc'}/s_c s_{c'} \\ &= (s_{ac'} + s_{bc'})/s_c s_{c'} \\ &= (s_a s^*_{ac'} + s_b s^*_{bc'})/s_c \\ &\quad \text{(simply introducing cancelling terms)} \end{aligned}$$

and since $s_c = s_a + s_b$, the above expression resembles the Lance–Williams update formula for the average linkage (group average or UPGMA) method. The 'intramax' method used by Masser and Scheurwater[55] is similar but not identical to the foregoing: standardisation is firstly carried out on the asymmetric data; a symmetric matrix is obtained from this standardised matrix; following agglomeration of the two zones of maximum standardised symmetric flow, the initial asymmetric flow matrix is updated according to: $f_{cc'} = f_{ac'} + f_{bc'}$ and $f_{c'c} = f_{c'a} + f_{c'b}$, where $c = a \cup b$ and c' is any other region. The asymmetric flows matrix (now of row and column dimensions one less than formerly) is standardised as previously, made symmetric, and a further agglomeration carried out. In summary, whereas the algorithm of Domengès may be said to consist of the sequence of operations: make symmetric, standardise, and successively carry out all agglomerations, the algorithm of Masser and Scheurwater consists of: standardise, make symmetric, agglomerate, adjust original matrix, restandardise, make symmetric, agglomerate, and so on. Other alternative formulations of this algorithm have also been described by Masser and Brown[56] and Hirst.[57]

If required, it would be a trivial matter to incorporate a contiguity constraint into the directed single-linkage algorithm. In the case of the variant of the average linkage method proposed by Masser and Brown,[56] a contiguity constraint was included, but – the authors state – it was found to have no effect whatsoever on the resulting hierarchy. Barring possible computational advantages, it appears that the introduction of a contiguity constraint is not always necessary in the analysis of geographical interaction flow data, since most clusters (whether of the connected or compact type) are inherently contiguous.

5. CONCLUSION

For general-purpose constrained clustering, the algorithms described in this article fall into two broad classes: we may choose a centroid-based agglomerative algorithm, which attempts to construct compact clusters, but with such attendant difficulties as interpretability due to inversions or losing the given asymmetric character of dissimilarities; or, alternatively, we may use a single-linkage related algorithm, with the chaining disadvantage. These approaches might frequently be complementary since they have somewhat different objectives.

Two illustrative problems and algorithmic solutions, using the material in this article, are as follows.

A clear example of the contiguity-constrained clustering problem is the grouping of people/areas on the basis of some given set of socio-economic attributes. It might be expected that the objects of analysis which come from major urban areas would be grouped together. Consider now the presence of a contiguity constraint: the resulting clustering ought to clearly demarcate the urban areas, and instead group with them their respective hinterlands. The image which comes most readily to mind is one of modes or peaks. Thus the possibility of using techniques designed for mode seeking (Section 3), through the conversion of the attribute vectors into real-valued weights, or through adaptation of the density gradient dissimilarity.

For taking time into account in a clustering, a variance-based agglomerative approach (Section 2) with a contiguity matrix could be employed; so also could a mode-seeking approach (Section 3), where an asymmetric dissimilarity is defined between objects which are contiguous or adjacent in time.

Contiguity-constrained clustering has not been seen widely to date as an important development in clustering (a notable exception is Gordon).[58] In this article we have attempted to show that clustering of this kind has been implicitly used for some time in, amongst other areas, point pattern recognition. We have also attempted to show that a fundamental choice in contiguity-constrained clustering algorithms is between compactness criteria versus connectivity criteria (in this, we follow Fischer).[8] Finally we have attempted to lay out recent work in a framework which, we hope, will facilitate the practical choice of appropriate algorithms, and lead to further theoretical and empirical results in this area.

REFERENCES

1. A. D. Gordon, Classification in the presence of constraints. *Biometrics* **29**, 821–827 (1973).
2. Y. Lechevallier, *Classification automatique optimale sous contrainte d'ordre total*, INRIA Report No. 200, Le Chesnay (1976).
3. A. D. Gordon, *Classification*, Chapman and Hall (1981).
4. R. M. Cormack, A review of classification. *Journal of the Royal Statistical Society* A **134**, 321–367 (1971).
5. M. Jambu and M. O. Lebeaux, *Classification automatique pour l'analyse des données*, Dunod, Paris (1978); *Cluster Analysis and Data Analysis*, North-Holland, Amsterdam (1983).
6. A. D. Gordon and H. J. B. Birks, Numerical methods in Quaternary Palaeoecology. II. Comparisons of pollen diagrams. *New Phytologist* **73**, 221–249 (1974).
7. P. Monestiez, Méthode de classification automatique sous contraintes spatiales. *Statistique et Analyse des Données* **2**, 75–84 (1977).
8. M. M. Fischer, Regional taxonomy. *Regional Science and Urban Economics* **10**, 503–537 (1980).
9. F. Murtagh, Complexities of hierarchic clustering algorithms: state of the art. *Computational Statistics Quarterly* **1**, 101–113 (1984).
10. D. Cheriton and R. E. Tarjan, Finding minimum spanning trees. *SIAM Journal of Computing* **5**, 724–742 (1976).
11. A. C. Yao, An $O(|E| \log \log |V|)$ algorithm for finding minimum spanning trees. *Information Processing Letters* **4**, 21–23 (1975).
12. F. J. Rohlf, Algorithm 76: hierarchical clustering using the minimum spanning tree. *The Computer Journal* **16**, 93–95 (1975).
13. A. Ferligoj and V. Batagelj, Clustering with relational constraint. *Psychometrika* **47**, 413–426 (1982).
14. R. Webster and P. A. Burrough, Computer-based soil mapping of small areas from sample data. II. Classification smoothing. *Journal of Soil Science* **23**, 222–234 (1972).
15. S. Openshaw, A regionalisation program for large data sets. *Computer Applications* **1**, 136–160 (1974).
16. A. Thauront, Application d'un problème de classification avec contrainte de contiguïté. *Consommation* **1**, 35–60 (1976).
17. L. Lebart, Programme d'agrégation avec contraintes. *Les Cahiers de l'Analyse des Données* III, 275–287 (1978).
18. C. Roche, Exemple de classification hiérarchique avec contraintes de contiguïté: le partage d'Aix-en-Provence en quartiers homogènes. *Les Cahiers de l'Analyse des Données* III, 289–305 (1978).
19. C. Perruchet, Classification sous contrainte de contiguïté continue. *Actes de la Société Francophone de Classification*, IRISA Report No. 162, Rennes, 192–207 (1982).
20. C. Perruchet, Constrained agglomerative hierarchical classification. *Pattern Recognition* **16**, 213–217 (1983).
21. F. Murtagh, Fondements théoriques de la classification hiérarchique sous contrainte de contiguïté continue. *Actes de la Société Francophone de Classification*, IRISA Report No. 162, Rennes, 177–191 (1982).
22. S. Openshaw, A geographical solution to scale and aggregation problems in region-building, partitioning and spatial modelling. *Transactions of the Institute of British Geographers, n.s.* **2**, 459–472 (1977).
23. C. T. Zahn, Graph-theoretical methods for detecting and describing Gestalt clusters. *IEEE Transactions on Computers* C-20, 68–86 (1971).
24. J. R. Slagle, C. L. Chang and R. C. T. Lee, Experiments with some cluster analysis algorithms. *Pattern Recognition* **6**, 181–187 (1974).
25. J. R. Slagle, C. L. Chang and S. R. Heller, A clustering and data-reorganizing algorithm. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-15, 125–128 (1975).
26. R. C. T. Lee, Clustering analysis and its applications. In *Advances in Information Systems Science*, edited J. T. Tou, vol. 8, pp. 169–292, Plenum Press, New York (1981).
27. R. Urquhart, Graph theoretical clustering based on limited neighbourhood sets. *Pattern Recognition* **15**, 173–187 (1982); erratum, *Pattern Recognition* **15**, 427 (1982).
28. R. Sibson, The Dirichlet tesselation as an aid in data analysis. *Scandinavian Journal of Statistics* **7**, 14–20 (1980).
29. N. Ahuja, Dot pattern processing using Voronoi neighbourhoods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-4, 336–343 (1982).
30. J. Fairfield, Segmenting dot patterns by Voronoi diagram concavity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-5, 104–110 (1983).
31. E. Shaffer, R. Dubes and A. K. Jain, Single-link characteristics of a mode-seeking clustering algorithm. *Pattern Recognition* **11**, 65–70 (1979).
32. W. L. G. Koontz, P. M. Narendra and K. Fukunaga, A graph-theoretic approach to nonparametric cluster analysis. *IEEE Transactions on Computers* C-25, 936–944 (1976).
33. B. Johnston, T. Bailey and R. Dubes, A variation on a nonparametric clustering method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1, 400–408 (1979).
34. R. Mizoguchi and M. Shimura, A nonparametric algorithm for detecting clusters using hierarchical structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-2, 292–300 (1980).
35. D. Wishart, Mode analysis: a generalization of nearest neighbour which reduces chaining effects. In *Numerical Taxonomy*, edited A. J. Cole, Academic Press, London, pp. 272–281 (1969).
36. J. O. Katz and F. J. Rohlf, Function-point cluster analysis. *Systematic Zoology* **22**, 295–301 (1973).
37. M. P. van Oeffelen and P. G. Vos, An algorithm for pattern description on the level of relative proximity. *Pattern Recognition* **16**, 341–348 (1983).
38. J. A. Hartigan, *Clustering Algorithms*, Wiley, New York (1975).
39. P. M. Narendra and M. Goldberg, Image segmentation with directed trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-2, 185–191 (1980).
40. R. M. Haralick and I. Dinstein, A spatial clustering procedure for multi-image data. *IEEE Transactions on Circuits and Systems* CAS-22, 440–450 (1975).
41. M. Goldberg and S. Shlien, A clustering scheme for multispectral images. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-8, 86–92 (1978).
42. J. Kittler, A locally sensitive method for cluster analysis. *Pattern Recognition* **8**, 23–33 (1976).
43. J. Kittler, Comments on 'Single-link characteristics of mode-seeking clustering algorithm'. *Pattern Recognition* **11**, 71–73. 1979).
44. P. M. Narendra and M. Goldberg, A non-parametric clustering scheme for LANDSAT. *Pattern Recognition* **9**, 207–215 (1977).
45. S. W. Wharton, A contextual classification method for recognizing land use patterns in high resolution remotely sensed data. *Pattern Recognition* **15**, 317–324 (1982).
46. S. W. Wharton, A generalized histogram clustering scheme for multi-dimensional image data. *Pattern Recognition* **16**, 193–199 (1983).
47. K. Ozawa, CLASSIC: a hierarchical clustering algorithm based on asymmetric similarities. *Pattern Recognition* **16**, 201–211 (1983).
48. P. B. Slater, Combinatorial procedures for structuring internal migration and other transaction flows. *Quality and Quantity* **15**, 179–202 (1981).

49. L. Hubert, Min and max hierarchical clustering using asymmetric similarity measures. *Psychometrika* **38**, 63–72 (1973).
50. A. Findlay and P. B. Slater, Functional regionalization of spatial interaction data: a comment. *Environment and Planning* A **13**, 645–646 (1981).
51. P. B. Slater, Comparisons of aggregation procedures for interaction data: an illustration using a college student international flow table. *Socio-Economic Planning Sciences* **15**, 1–8 (1981).
52. J. P. Boyd, Asymmetric clusters of internal migration regions of France. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-10, 101–105 (1980).
53. R. E. Tarjan, An improved algorithm for hierarchical clustering using strong components. *Information Processing Letters* **17**, 37–41 (1983).
54. D. Domengès, Classification ascendante hiérarchique d'après un critère adapté aux tableaux de flux. *Les Cahiers de l'Analyse des Données* VII, 169–172 (1982).
55. I. Masser and J. Scheurwater, Functional regionalization of spatial interaction data: an evaluation of some suggested strategies. *Environment and Planning* A **12**, 1357–1382 (1980).
56. I. Masser and P. J. B. Brown, Hierarchical aggregation procedures for interaction data. *Environment and Planning* A **7**, 509–523 (1975).
57. M. A. Hirst, Hierarchical aggregation procedures for interaction data: a comment. *Environment and Planning* A **9**, 99–103 (1977).
58. A. D. Gordon, Methods of constrained classification. In *Analyse des Données et Informatique*, edited R. Tomassone, INRIA, Le Chesnay, pp. 161–171 (1980).