M.Tech. (Computer Science) Dissertation Series

ANALYSIS OF CLUSTER VALIDITY PROBLEM

a dissertation submitted in partial fulfilment of the
requirements for the M. Tech. (Computer Science)
degree of the Indian Statistical Institute

by

Girish Kumar Gupta

under the supervision of

Dr. C. A. Murthy

INDIAN STATISTICAL INSTITUTE
203, Barrackpore Trunk Road
Calcutta - 700 035

ACKNOWLEDGEMENT

I take this oppurtunity to express my gratitude towards my guide, Dr. C. A. Murthy for his valuable guidance and all-out help throughout the course of my work. Thanks are also due to the National Centre for Knowledge Based Computing Systems for providing me with the computing facilities, and to all my teachers and friends for their help and cooperation.

ABSTRACT

This work deals with the cluster Validity problem. Here three new Validation indices have been defined. These indices have been calculated for various clustering techniques. Concepts in Computational geometry like Voronoi diagram, Delaunay triangulation and Gabriel graph have been utilized for the purpose.

Contents

Chapter 1

1. Introduction

Cluster analysis deals with automating a natural and commonly utilized human activity of forming classes or groups of similar objects, irrespective of their origin.Thus the objects to be clustered could be patients in a hospital, different brands of a consumer product, students at a university, or pixels in a digital image. It may not be possible to say exactly what a cluster is in abstract terms, without a statement of the criterion and the implementing algorithm. Cluster analysis plays an important role in solving many problems in pattern recognition and image processing. It is used for feature selection , in numerous applications involving unsupervised learning (where it is difficult to assign a reliable category label to the training patterns), and speech and speaker recognition. But a novice user of cluster analysis soon finds that though the intuitive idea of clustering is clear enough, the details of actually carrying out such an analysis entails a host of problems. Anderberg [5] has proposed an outline of the major steps of a cluster analysis. The major steps involved are the choice of the data units, the choice of the variables and the similarity measure, decisions regarding what to cluster, the clustering criterion, and the algorithm to be used.

Once we obtain the clusters the results are to be suitably interpreted. At the least level of sophistication, the clusters are summary statistics like the mean and the variance. At the next level, the set of choices used for cluster analysis are so well defined that the clusters necessarily have the desired propeties. And at the highest level, the results of cluster analysis are an aid to reasoning from the data to the explanatory hypothesis about the data.

Any given set of data may admit of many different but meaningful classifications. Each classification may pertain to a different aspect of the data. Cluster analysis is a device for suggesting hypothesis. The classification of data units or variables obtained from a cluster analysis procedure has no inherent validity. The analyst should not feel any pressure to embrace a particular classification, nor should he feel bound to the details of a classification he finds interesting. The worth of a classification and its underlying explanatory structure is to be justified by its consistency with known facts and without regard to the manner of its original generation. A set of clusters is not itself a finished result, but only a possible outline.

There are many clustering algorithms found in the literature. They involve fixed sequence of operations which systematically ignore some aspect of structure, while intensively dwelling on others. So to a considerable extent a set of clusters reflects the degree to which the data set conforms to the structural forms embedded in the clustering algorithm.

Clustering algorithms tend to generate clusters even when applied to random data. If one has a great deal of experience with a particular clustering method, and some prior information about the data being clustered, the results of a cluster analysis can confirm or deny assertions about the data and suggest subsequent analysis. However, the user of a clustering algorithm is often unsure about the data and has little experience with a particular type of data or a particular clustering method. Lack of information about the data is often the motivation for clustering the data in the first place. In this case the user searches for objective meaning and needs quantitative measures of significance for evaluating clustering structures. Cluster validaton refers to procedures that evaluate the results of cluster analysis in a quantitative and objective fashion. The task of cluster Validity is to separate the artifacts from the structure. The fundamental questions we address can be stated from user's point of view as : Are the clusters and the structures generated by a clustering method or algorithm significant enough to provide evidence for hypothesis about the phenomenon being studied. The problem can be viewed as in fig. 1.1 on the next page.

Data are first checked for clustering tendency, and only if the data tend to be non-random is clustering attempted. The validation process judges the success of an algorithm in imposing a structure as well as the suitability of the structure for the data. We assume that the data may be given in the form of a proximity matrix or a pattern matrix. And the imposed structure may be partitional, or hierarchical, depending on the clustering methodology used.

The validation of the results of imposing a structure on data with a clustering method may be done using the following criterion :-

1. **Compactness criterion** : measures the inner strength, or concentration or cohesion or uniqueness of an individual cluster with respect to its environment.

2. **Isolation criterion** : measures the separation or gap between a cluster and its environment.

3. **Global fit criterion** : measures the accuracy with which the structure describes the relationships between clusters, as well as the extent to which all the clusters are individually valid.

4. **Intrinsic dimensionality criterion** : determines the "shape" of a cluster and provides information about representing the patterns in a cluster.

There is a need to establish a methodology whereby one can incorporate specific criteria into a program and the type of imposed stucture. This problem is attacked by fixing

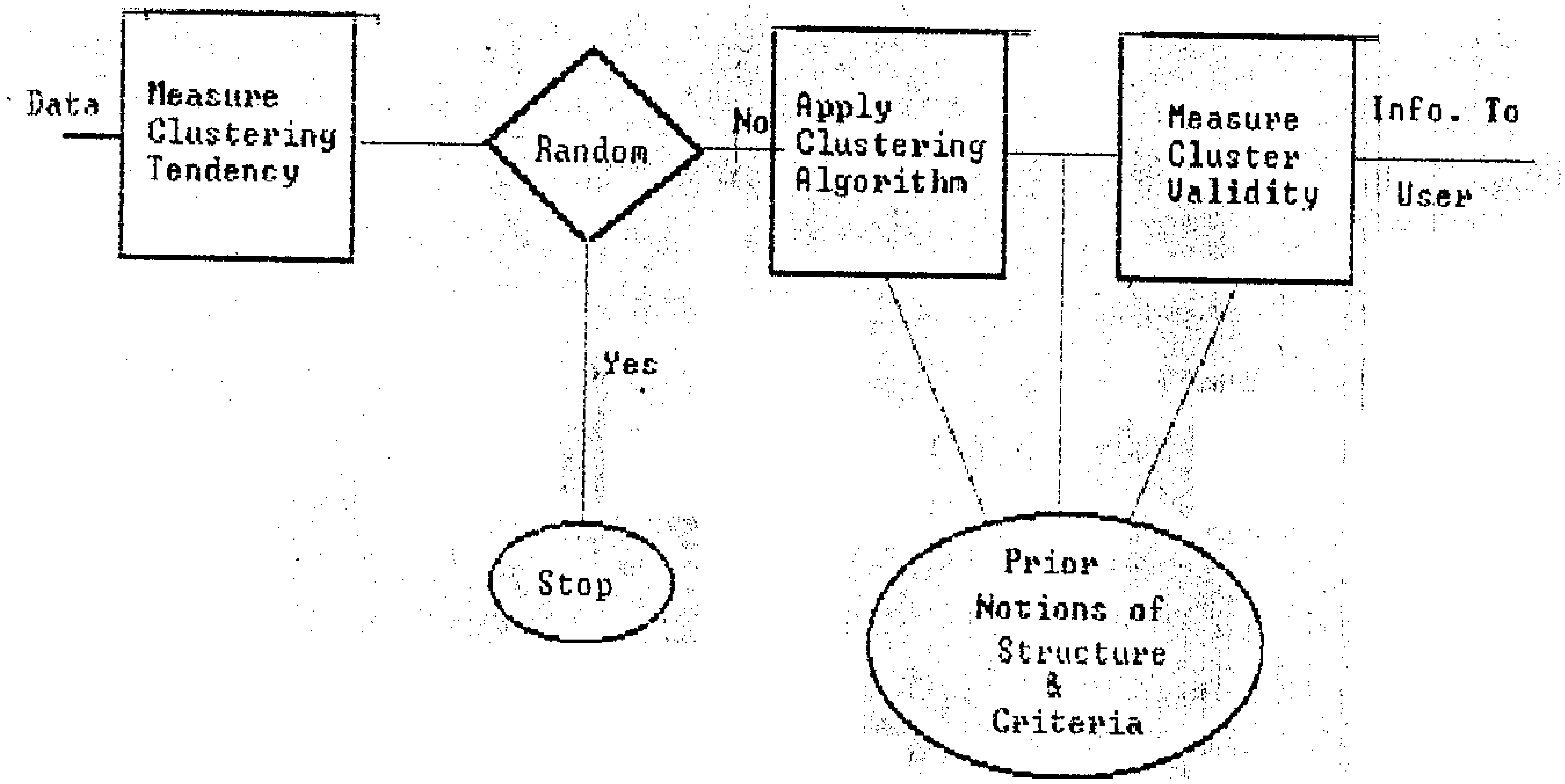

Data
Measure Clustering Tendency
Random
No
Yes
Stop
Apply Clustering Algorithm
Measure Cluster Validity
Info. To User
Prior Notions of Structure & Criteria

Fig. 1.1 : Scope of Cluster Validity Problem

the data type and the type of imposed structure.The following factors are helpful in deciding which of the above criteria/criterions to choose from, in order to validate the clusters :-

1. **Null hypothesis** : This is a hypothesis about the meaning of "no clustering", often expressed as a concept of randomness or the antithesis of clustering. There are only two hypothesis that have been studied in the literature. They are the **Random Graph Hypothesis** and the **Random Position Hypothesis.**The first is applicable in studies involving symmetric proximity matrices whose entries are rank orders. For eg., an n x n ordinal dissimilarity matrix has zeros in the diagonal and the numbers 1, 2, ... n(n-1)/2 in the upper triangle without ties : the most familiar pair of data items has rank 1. The Random Graph Hypothesis is that all [n(n-1)/2]! such matrices are equally likely. The Random Position Hypothesis views the n patterns as independent samples from a d dimensional distribution.

2. **The ideal cluster** : This factor establishes the user's prior concept of what a cluster means and sets the goal for a clustering method. One can formulate an idea of cluster from an assumed mathematical model or from prior work in the subject matter.

3. **Sample size** : Increasing the number of patterns can increase the confidence in a particular structure, but may increase the computational burden.

4. **Details of imposed structure** : Each clustering method imposes its own set of restrictions. Sometimes the restrictions are implicit in the definition of the clusters. Many clustering algorithms always finds clusters which are ball shaped. Others always place patterns which are closest in the same cluster.

1.1 Survey of literature

The first step in carrying out the study of cluster validity is to verify whether the data set shows a clustering tendency. Various strategies are found in the literature to measure this tendency. Some approaches require a rank order dissimilarity matrix $[r_{ij}]$ and is based on the concept of a random graph. We begin with a set of n nodes, one per pattern. A threshold graph G(n,e) is is an undirected graph containing n nodes and e edges with edge (i,j) being entered if $r_{ij} \leq e$. Under the Random Graph Hypothesis, these edges are entered randomly.

The first test requires the number of edges, E needed to connect a random graph. Knowing the distribution of the number of edges required to connect a random graph, permits one to judge how many edges must be added before deciding that the data are random. Ling and Killough [9] produced an exact equation for the probability, based on the results of Riddel and Uhlenbeck [13]. Ling [1] adopts the Random Graph Hypothesis, which requires all [n(n-1)/2]! ordinal dissimilarity matrices to be equally likely.

The distribution function for E is denoted by P(n,e) = Prob($E \leq \frac{e}{n}$)

If e^* is observed in a particular situation as the level at which the graph for the data being studied first becomes connected, or the level at which all data items are absorbed into the same single link cluster, then the clustering tendency is tested as follows :

If P(n,e) $\leq$ 0.99, evidence exists for the conclusion that the dissimilarity matrix was not chosen at random. The threshold 0.99 is arbitrary. The intuitive idea behind this is that within-cluster edges tend to occur before the between cluster edges when the data are clustered, thus delaying the formation of a connected graph. There are other tests based on the distribution of node degrees, and the number of nodes in clusters. The details can be found in the literature [12,2] .

Once the clustering tendency of the data is verified, the next step is to find out how well does a hierarchy fit a proximity matrix, or whether the partition obtained from a pattern matrix is valid, or whether the individual clusters appearing in a hierarchy are valid. We refer to the literature [3,4,15,14] for the various methods adopted for Cluster-Validity.

1.2 Our Approach

Most of the cluster Validity techniques employ various statistical tools to define the actual test of Cluster Validity. We are not following that approach. We assume that our samples come from a population of known classification, and that the classes are non-overlapping. We have then applied the clustering techniques and found the misclassified samples/misclassified areas. These misclassifications provide an idea of the Validity of the method.

For various sample sizes and for various data sets the averages of the Validity Indices that we define in the next Chapter have been calculated. The Validity-Indices have been defined in such a way that they take the value 1, if the classification is correct. The aim of this work is to find the average of Validity Indices for various clustering methods, and for various data sets so that one can get an idea of the Validity of the technique for unknown data sets. Expected values (Means) of the Validity-Indices are better measures of Validation, though they are difficult to calculate. We are not considering such calculations in our work.

It may be mentioned that though a lot is known about the various clustering algorithms, a good literature on the merits and demerits of various algorithms and their applicability to specific data sets is absent from the literature. We hope that the results given at the end will throw light on the performance of various clustering methods. This knowledge can be of immense help while applying the method of cluster analysis to solve real life problems.

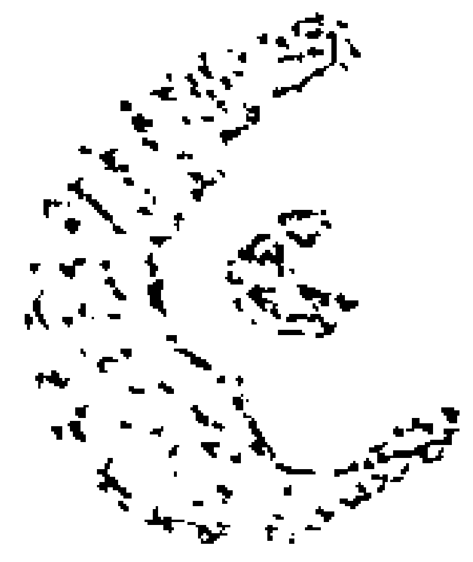
Chapter 2

2. Definition of new Validity Indices

To judge whether an algorithm gives the right clusters, one should know the original classification. Thus an approach to cluster Validity is to apply the algorithm to data sets with known classification and check the results. To get a quantitive idea of the results of such a comparision, and of the clustering methodology for the polulation in general, we define various Validity Indices.In defining these Validity Indices, we have assumed that our samples come from a population of known classification, and that the clusters are non-overlapping. We have then drawn random samples from a population with known classification, and proceeded to find the various Validity Indices that we define below. Observe that any Index that is found is dependent on the samples, and therefore different Indices will be found for different samples of the same size. This necessitates the knowledge of the probability of occurence of different sample sets. In this project the calculation of this probability is not being tackled. After the calculation of these probalities, and suitably incorporating them with the Validity Indices for the respective samples, this process results in Validation of the given procedure for the given population. Detailed studies for the validation of different clustering techniques for different data sets is absent in the literature, though some observations about the application of some techniques to some data sets are known. We have defined three Validity Indices in carrying out our study. The Indices have been defined such that they lie between 0 and 1, and higher the value of these Indices, closer are the clusters to the original population cluster. Hence the Validity Index is 1 for the right classification. We define the Validity Indices, VI_1, VI_2 & VI_3 in the following sections.

2.1 Definition of VI_1

The first Validity Index (VI_1) that we define evaluates on the assumption that, in general, the performance of a clustering algorithm improves if the size of the sample is increased. This result, however, doesnot apply to all clustering algorithms. K-means algorithm, for example, will not correctly classify the following type of data sets,however large the sample size may be.



But some other algorithms, like the set-estimation method (to be discussed in the next chapter) and the single linkage algorithm, guarantees that we always get the right classification if the sample size is very large.Now, for the given data set if the derived classification is found to be incorrect, the proposed method of Validity appends some more points to the given data set to achieve the right classification. The addition of these points may be incorporated subjectly, or randomly. Both methods of adding points will give different Validity Indices. We have selected the second option, and have not considered the computation of the probabilities. We now define VI_1 as given below :

$$VI_1 = \frac{1}{(1 + \frac{\text{N-Added}}{N})} \tag{2.1}$$

where,

N-Added = Number of points added to get the right classification, and

N = Size of the original sample.

It can be easily seen that $VI_1 = 1$, for the correct classification, and VI_1 goes to 0, when N-Added→ ∞.

2.1.1 Computation of VI_1

To compute VI_1, all we need to know is the number of points to be added to the given sample, so that we get the right classification. We are taking the points to be added randomly from the population from which the sample has come. We apply the clustering algorithm to the given sample, and verify whether we have got the right classification. We stop if we get the right classification, or when we have added MAXLIMIT number of points. The number of points added upto this stage is used to compute VI_1. Otherwise, we add one more point to the data set, and the above process is repeated.

2.2 Definition of VI_2

The problem with VI_1 is that it is computationally very expensive, and there is no guarantee that we will finally get the right classification. So we have defined another Validity Index (VI_2) which can be computed by a a single run of the program, while giving a fair idea of the Validity of the clusters obtained. Let,

a_{ij} = Number of points of i^{th} cluster going to j^{th} cluster, and

k = Number of clusters

then

$$VI_2 = 1 - \frac{\left(\sum_{i=1}^{k} \left(\sum_{j=1,i\neq j}^{j=k} \frac{a_{ij}}{\text{No. of points in } i^{th} \text{ cluster}}\right)\right)}{k-1} \tag{2.2}$$

Again we find that

$$(VI_1)_{min} = 1, \text{and}$$

$$(VI_2)_{max} = 0$$

2.2.1 Computation of VI_2

The computation of VI_2 is also straightforward. Here we need to know the number of misclassified points (a_{ij}, j = 1...k) of each cluster C_i. This we find by comparing the clusters obtained with the original population cluster, and finding the number of misclassified points.

While comparing the clusters we have to be careful about the label we assign to each cluster of the sample. We have chosen to perform the comparision for each possible labelling of the sample, and choosing the one which gives the minimum number of misclassified points as the correct labelling.

2.3 Definition of VI_3

The Validity Indices VI_1 & VI_2 are somewhat sensitive to the size of the samples, and so we define another Index of Validity (VI_3), which is based on the computation of the areas of the clusters. The areas are defined below. Let

- a_i = The area corresponponding to cluster C_i where the points are rightly classified,

- A_i = The area corresponding to a_i when all points in cluster C_i are classified correctly,and
- k = Number of clusters.

VI_3 is now defined as

$$VI_3 = \frac{\sum_{i=1}^{k} \frac{a_i}{A_i}}{k} \qquad (2.3)$$

2.3.1 Computation of VI_3

To compute VI_3 we need to compute the areas a_i and A_i for each cluster C_i. The points needed to compute A_i is known from the original population cluster. The points needed to compute a_i are the points which are present in the corresponding clusters of both the data set and the given population cluster. Now all that remains to be computed are the above areas from a given set of points. The computation of the above areas is not an easy task. We have found a way to compute the above areas, using the Voronoi diagram, the details of which are given in Chapter 4.

Chapter 3

3. Clustering Methods Studied

In this chapter, we describe in brief, various algorithms that we have considered for our study. All the algorithms considered are valid only for non-overlapping clusters only.

3.1 K-means algorithm

This is one of the most popular clustering techniques, because of its advantage in terms of space and time complexity, and ease of implementation.It is based on the minimization of a performance index J, which is the sum of squared distances from all points in a cluster domain to the cluster centre.

$$J = \sum_{j=1}^{k} \sum_{x \in C_j} \|x - m_j\|^2$$

where,

k = No. of clusters,

C_j = Set of samples belonging to the j^{th} cluster,and

$$m_j = \frac{1}{N_j} \sum_{x \in C_j} x$$

m_j is the sample-mean of C_j, and N_j represents the number of samples in C_j. The algorithm can be found in [10].

Although, no general proof of convergence exists for this algorithm, it can be expected to yield acceptable results when the data exhibits characteristic pockets which are relatively far from each other. This algorithm assumes that the number of clusters be known apriori. The selection of the wrong seed points, may in some cases, may make all the difference between a wrong classification, and a correct classification. We have followed the following method for selecting the seed points :

Let

S = $\{x_1, x_2 \ldots x_n\}$ be the given data set,

k = No. of clusters, and

$$t = \lfloor \frac{n}{k+1} \rfloor$$

The seed points are then given by $\{x_t, x_{2t}, x_{3t} \ldots x_{kt}\}$

3.2 Single-linkage Algorithm

This is a hierarchical clustering algorithm, wherein we start with n clusters and at each stage reduce the number of clusters by one. Let us first define

$$d(A, y) = \inf_{x \in A} d(x, y)$$

$$d(A, B) = \inf_{x \in A, y \in B} d(x, y)$$

where,

A and B are the data sets.

At stage i, there are (n-i) clusters. We merge two clusters C_k and C_l for which $d(C_k, C_l)$ is minimum among all the clusters at this stage. This process is repeated for i=0,1...(n-1). So we get a dendogram. The number of clusters is now decided using some heuristic which ensures that the step jump between two successive levels in the dendogram is significant. The dendogram is now cut at this level, and the disconnected components in the dendogram are the clusters.

In carrying out our study, we have assumed that the number of clusters is known apriori. This ensures that we are not very much biased when we compare the results of this algorithm with other algorithms, like the K-means algorithm. This algorithm is known to give good results when the cluster consists of long chain like structures.

3.3 Complete-linkage Algorithm

The steps followed by this algorithm are exactly the same as that followed by the single-linkage algorithm, except that the dissimilarity measure d is redifined here as

$$d(A, y) = \sup_{x \in A} d(x, y)$$

$$d(A, B) = \sup_{x \in A, y \in B} d(x, y)$$

where,

A and B are the data sets.

This algorithm gives results which are comparable to that obtained by the K-means algorithm, when the clusters are compact. The advantage here is that we need not know the number of clusters apriori, and the problem of seed selection as in the case of K-means algorithm is also not there.

3.4 Set-Estimation Method

This algorithm which was proposed by C.A. Murthy and D. Dutta Mazumdar will always provide the right classification if the number of points is very large, and the clusters are non-overlapping [6,7,8]. The major steps of this algorithm are given below :

Let S = $\{x_1, x_2 \ldots x_n\} \subseteq \Re^d$ be the given data set.

1. Find out the Minimum-spanning tree(MST) of S, where the edge-weight is taken as the Euclidean distance.
2. Calculate the sum of the edge-weights of the MST and call it l_n.
3. Let $h_n = \sqrt{\frac{l_n}{n}}$
4. Remove form the MST those edges whose edge-weights is less than $2h_n$. The connected components that we get are the clusters.

This method also doesnot require the apriori knowledge of the number of clusters. If the number of clusters that we get using this method is the same as that we obtain using the single-linkage method, then the clusters we obtain in both cases are identical.

Chapter 4

4. Computation of VI_3

We have seen in Chapter 2 that the computation of VI_3 requires the knowledge of the area of a cluster. The various steps involved in it are discussed here. The area can be calculated once the shape of the clusters is known.

The shape of a set of points has been defined in various ways. One common way to define the shape of a finite set of two dimensional points is its convex hull. But in many cases the underlying shape from which the points emerge is not convex. Edelsbrunner et al. [17] extended one definition of the convex hull and proposed a general definition of the shape (convex or otherwise) of a finite planar set. This is called **α-shape**. The **α-shapes** seem to capture the intuitive notions of the 'fine-shape' and 'crude-shape' of point sets. They are also subgraphs of the closest point or further point Delaunay triangulation. But the main drawback of this method is the dependence of the shape on the parameter α $(0 \leq \alpha \leq 1)$, and this α has to be chosen by trial and error. Our method of obtaining the shape of a set of points S, while capturing the intuitive notion of the shape of the set of points, is also free from this drawback.

Let us assume that we require to compute the area A (Cover set of points) for the set of points S. The major steps involved are given below:

1. Construct the Voronoi- diagram for the set of points S,
2. Construct the covering polygon of the set of points S (Cover Set of S),
3. Triangulate the above polygon and find out the required area A.

The details are now given in subsequent sections.

4.1 Voronoi diagram

Definition : Given a of N points S in the plane, we divide the plane into N regions such that for each point p_i in S, the region i is the locus of points (x,y) that are closer to p_i than to any other point of S. Such a partition defines what we call as the Voronoi diagram. We denote the Voronoi diagram by Vor(S), and the Voronoi polygon associated with p_i is denoted by V(i). The vertices of the Voronoi diagram are the Voronoi-vertices, and its line segments are the Voronoi edges.

We now give below, without proof, some of the properties of the Voronoi diagram. For proof, the reader is referred to [11].

- **P1:** Every Voronoi vertex in the Voronoi diagram is the intersection of at least three Voronoi edges.
- **P2:** For every Voronoi vertex, let C(v) denote the circle with centre at v, and passing through three or more points satisfying property P1. Then for every vertex v of the Voronoi diagram, the circle C(v) contains no other point of S.
- **P3:** Every nearest neighbour of p_i defines an edge of the Voronoi polygon V(i).

- **P4:** Polygon V(i) is unbounded iff p_i is a point in the boundary of the convex-hull of S.

- **P5:** The straight line dual of the Voronoi diagram is the triangulation of S, known as the Delaunay Triangulation.

4.1.1 Constructing the Voronoi diagram

By constructing the Voronoi diagram Vor(S), we shall mean to produce a description of the diagram as a planar graph embedded in the plane, consisting of the following items :

1. The coordinates of the Voronoi vertices,

2. The set of edges and the four edges that are their counterclockwise and clockwise successors at each extreme point. This implicitly provides a cycle around each vertex and around each face, in either direction.

We now give below two results we have use of in our algorithm. The proof can be referred in [11].

- **Result 1**

  Given a partition $\{S_1,S_2\}$ of S, let $\sigma(S_1, S_2)$ denote the set of Voronoi edges that are shared by pairs of polygons V(i) and V(j) of Vor(S), for $p_i \in S_1$ and $p_j \in S_2$. Then if S_1 and S_2 are linearly separated, then $\sigma(S_1, S_2)$ consists of a single monotone chain.

- **Result 2**

  If S_1 and S_2 are linearly separated by a vertical line with S_1 to the left of S_2, and σ cuts the plane into a left portion π_L and a right portion π_R, then the Voronoi diagram Vor(S) is the union of Vor(S_1) $\cap \Pi_L$ and Vor(S_2) $\cap \Pi_R$.

The algorithm, therefore, runs as follows :

1. Partition S into two subsets S_1 and S_2 of approx. equal size by median x-coordinate.
2. Construct Vor(S_1) and Vor(S_2) recursively.
3. Construct the dividing chain σ separating S_1 and S_2.
4. Discard all edges of S_2 that lie to the left of σ and all edges of Vor(S_1) that lie to the right of σ. The result is Vor(S), the Voronoi diagram of the entire set.

The main step of the above algorithm is the construction of the dividing chain σ. This can be done in linear time by following the algorithm given in [11].Hence the construction of Vor(S) can be carried out in O(nlog_2n) time, which is optimal.We have implemented the above algorithm on the assumption that not more than three points are co-circular.

4.2 Cover Polygon

Our next objective is to find the cover-polygon from the Voronoi-diagram. This will give us the shape of the set of points S.The following routines are devised for this purpose.

- *1.IsIntersects(e,v_1,v_2)* : This routines returns TRUE, if the line segment joining v_1 and v_2 intersects with the edge e. Otherwise, it returns FALSE.
- *2.ReverseVertex(v_1, e)* : This routine returns the vertex v_2 on the side of e, which is opposite to v_1.

To construct the cover-polygon, we move around the set of points in the counter-clockwise direction, and join the successive vertices which form part of our polygon. How these vertices are selected will be given in the steps that follow.

```
Step 1: Select the leftmost point of S as the intitial hull-vertex, and call it HullRoot.Let
            $h_1$ = HullRoot;
            e = Open edge of V($h_1$);
            $h_2$ = ReverseVertex($h_1$, e);

Step 2: repeat
    begin
        if (IsIntersects(e,$h_1$,$h_2$) == TRUE)
        begin
            Include the edge ($h_1$,$h_2$) in the cover-polygon;
            $h_1 = h_2$;
        end
        e = The next edge that we come across, while moving in the clockwise
            direction about $h_1$, and starting from the edge e;
        $h_2$ = ReverseVertex($h_1$, e);
    end;
    while ($h_1 \neq$ HullRoot)
```

Figures 6.1 to 6.7 illustrates the above procedure. Note that this method, in its present form, will fail to give the shape of a cluster having a ring like structure, as shown in fig. 5.3. This occurs because the cluster boundary in this case is not continuos. So once the outer boundary is traversed, we must construct the inner boundary. This can done following the steps given above, provided we can locate one of the points belonging to the inner boundary. We have not tackled this problem. So our method works only when the cluster boundary is continuos, which is usually the case.

4.3 Area of a polygon

Our next objective is to obtain the area of the polygon, which is obtained by successive triangulations. Let P_n be a polynomial with n vertices $v_0, v_1, \ldots v_{n-1}$ connected in that order. We now give below a recursive procedure **POLY-AREA**(P_n, v_i which returns the area of the polygon P_n. v_i is the reference vertex.

```
        Procedure POLY-AREA($P_n$ , $v_i$ )
begin
    if ( n = 3)
    begin
      Compute the of the triangle formed by the vertices ($v_0, v_1, v_2$).
      return area;
    end
    else
    begin
      Check whether the edge ($v_i, v_{i+1}$) lies inside the polynomial;
      if it lies inside,
      begin
        # Compute the area of the triangle formed by the vertices ($v_i, v_{i+1}, v_{i+2}$).
        # Remove the vertex $v_{i+1}$ from $P_n$, and add an edge ($v_i, v_{i+1}$)
        # Return (Area + POLY-AREA($P_{n-1}$, $v_{i+1}$))
      end
      else
        Return POLY-AREA($P_n$, $v_{i+1}$)
    end
end.
```

It may be noted that the edges of the Cover Polygon (CP) are a subset of the edges of the Gabriel-Graph(GG) and hence of Delaunay triangulation(DT) also. The Graph obtained from DT by removing those edges of DT which donot intersect its dual Voronoi-edge, is the Gabriel-Graph [16]. In constucting our polygon, we are including only those edges of GG, which lies on the periphery.

Chapter 5

5. Generation of data sets

In this chapter, we give a list of data-sets that we have generated for carrying out our study, alongwith other relevant details. We broadly divide our data-sets into three classes.

5.1 Data-Class I

The data-sets in this class consists of two unit squares separated by a distance δ as shown in fig. 5.1. We have selected the values of δ as $\delta = \{ 0.1; 0.2, 0.5, 0.8 \}$. We have considered the following distributions for our study:

- Uniform, and
- Triangular.

Ten data-sets for each value of δ and for each of the above distributions were considered. The study for each of these cases was then carried out for sample-size **N** ranging from 50 to 250, in steps of 50.

5.2 Data-Class II

The clusters in this Data-Class comprises of a concentric disc, and a ring. The shape of the clusters is shown in fig. 5.2. Let the radius of the disc be **r**, and the outer and inner radius of the ring be **R** and r_0 respectively. We choose the following value of the parameters :

- R = 1
- $r_0 = 0.8$
- Define δ-radius = r_0 - r

The value of the δ-radius was taken to be {0.3, 0.4, 0.5 }. The distribution is taken to be uniform. Ten data sets were generated for each δ, and the study was carried out for sample-size **N** ranging from 50 to 250, in steps of 50.

5.3 Data-Class III

This Data-Class consists of three clusters, arranged in the configuration shown in fig. 5.3.

The following two data-sets were considered under this Data-Class :

1. $\delta_1 = 0.8$, $\delta_2 = 0.5$, and
2. $\delta_1 = 0.5$, $\delta_2 = 0.8$.

The results are described in the next Chapter.

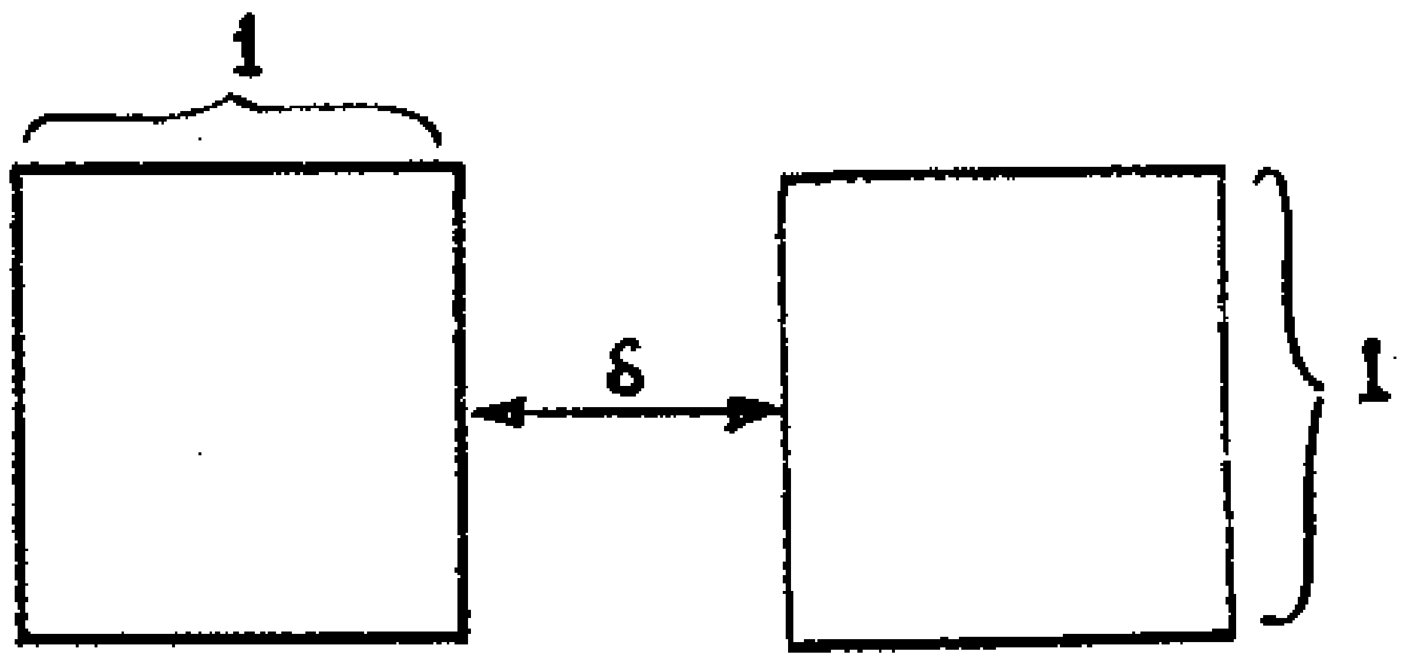


Fig. 5.1 : Data Class-I

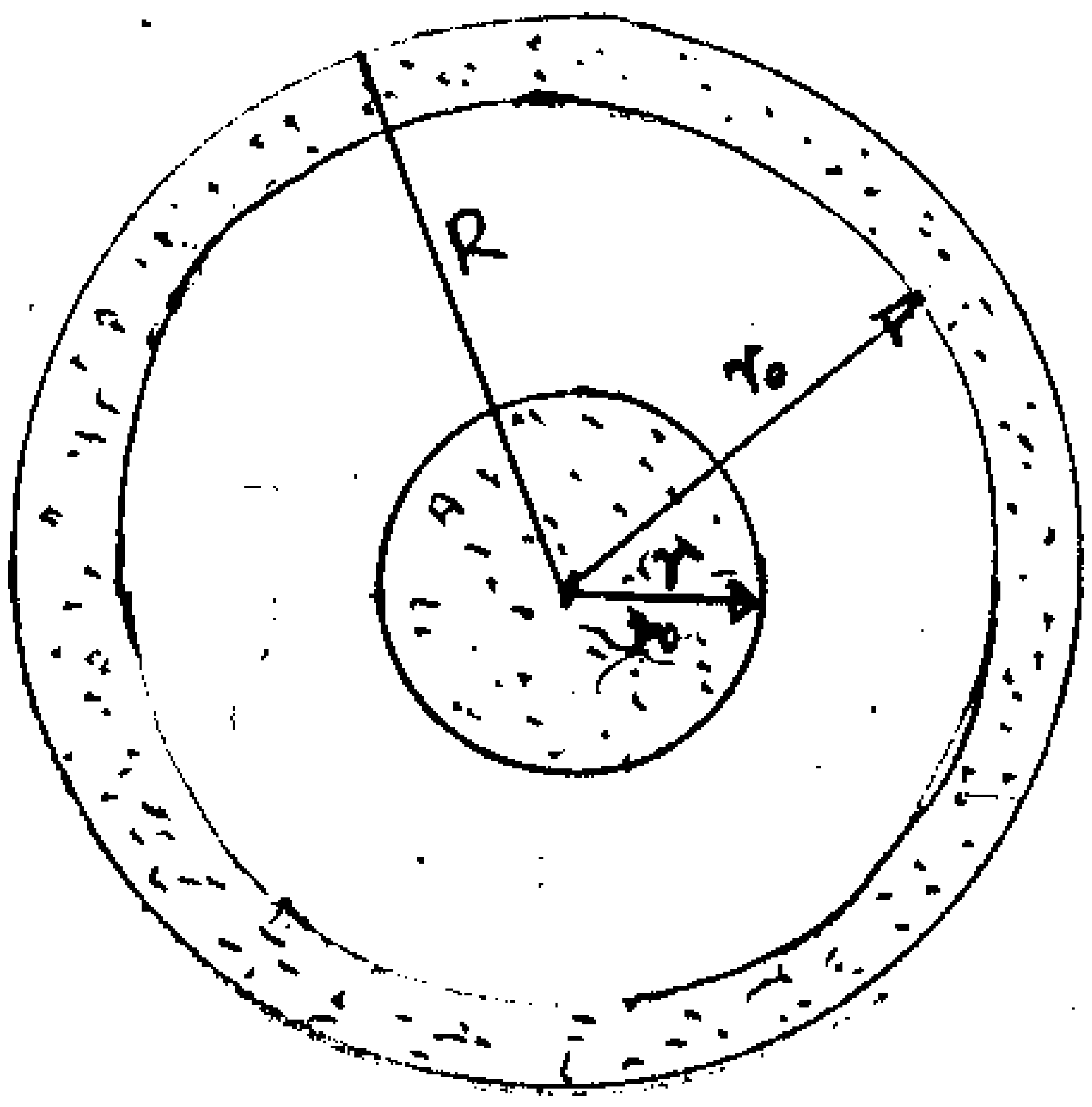


Fig. 5.2 : Data Class-II

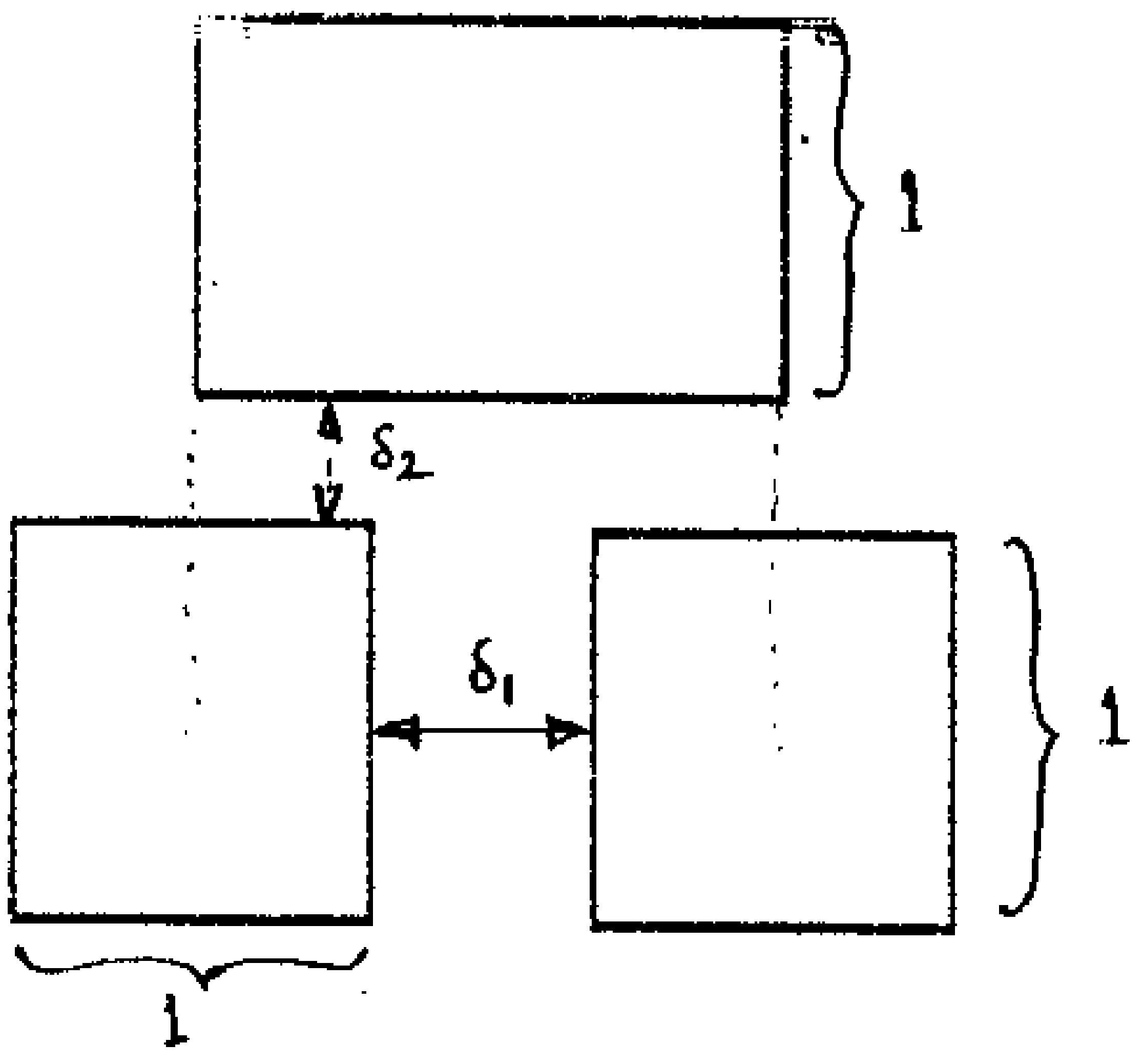


Fig. 5.3 : Data Class-III

Chapter 6

6. Results, Conclusions, and Scope for further Work

This chapter describes, in brief, some results we got during the course of our work, along with relevant observations and conclusions. We then give the scope for further work in the last section.

6.1 Results

6.1.1 Area of a cluster and VI_3

Before we give the results for the computation of Validity Indices, we cite an example to illustrate that the method followed by us actually gives the desired shape of a cluster. We have drawn random points from a semi-circular ring. We have then constructed the Cover-Polygon, from the Voronoi-diagram. The result is shown in fig. 6.1.

We also cite an example to show the computation of VI_3. The sample was taken for the Data-Class-I, and for $\delta = 0.1$ and n = 100. The clusters were obtained using the K-means algorithm. The computed areas are shown in figures 6.2, 6.3 and 6.4. The nomenclature used for labelling the areas can be referred in section 2.3. The computed value for VI_3 was found to be 0.7570.

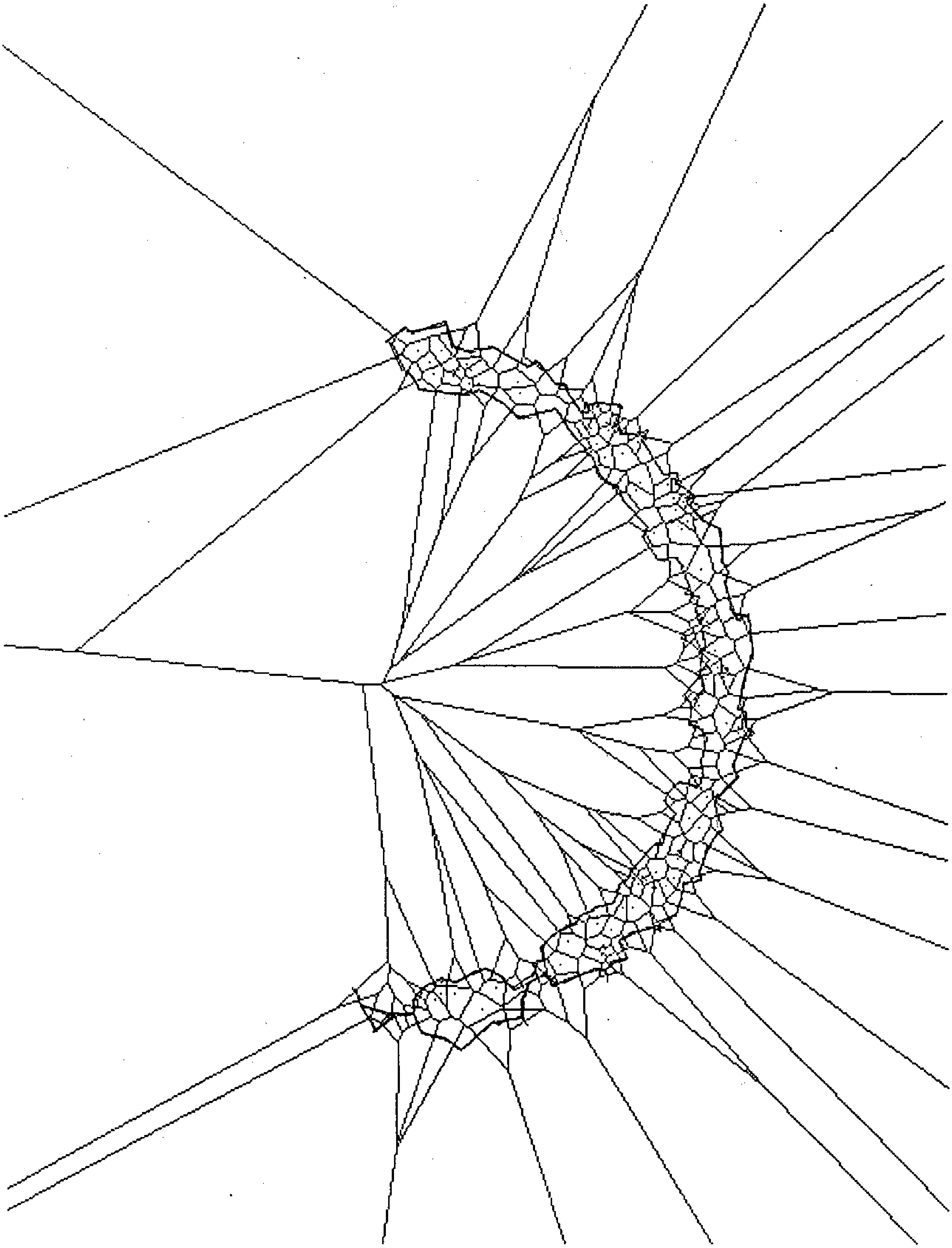


Fig. 6.1 : Shape of a set of points

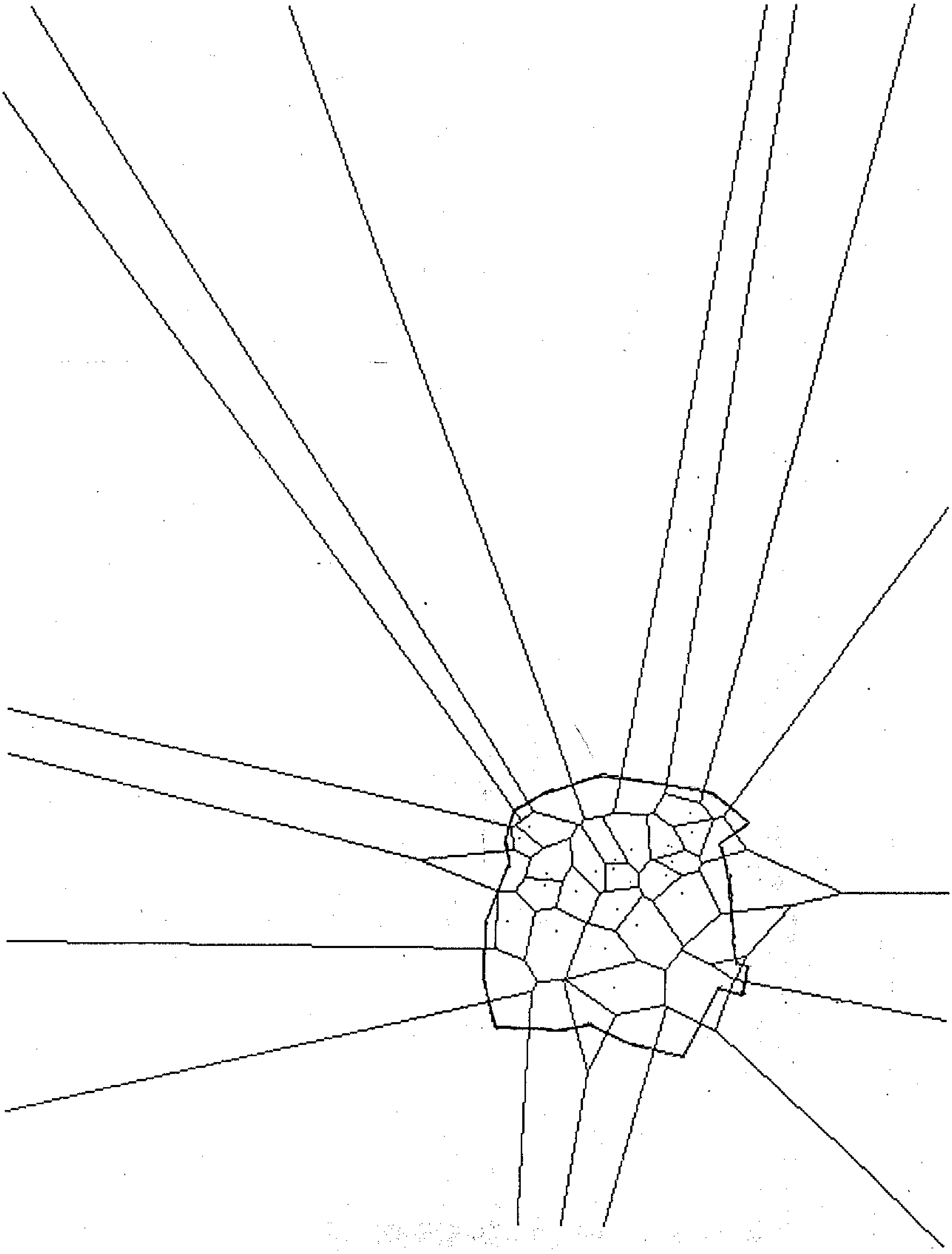


Fig. 6.2 : Area A_0/a_0 of cluster C_0

$A_0 = 0.6806$

$a_0 = 0.6806$

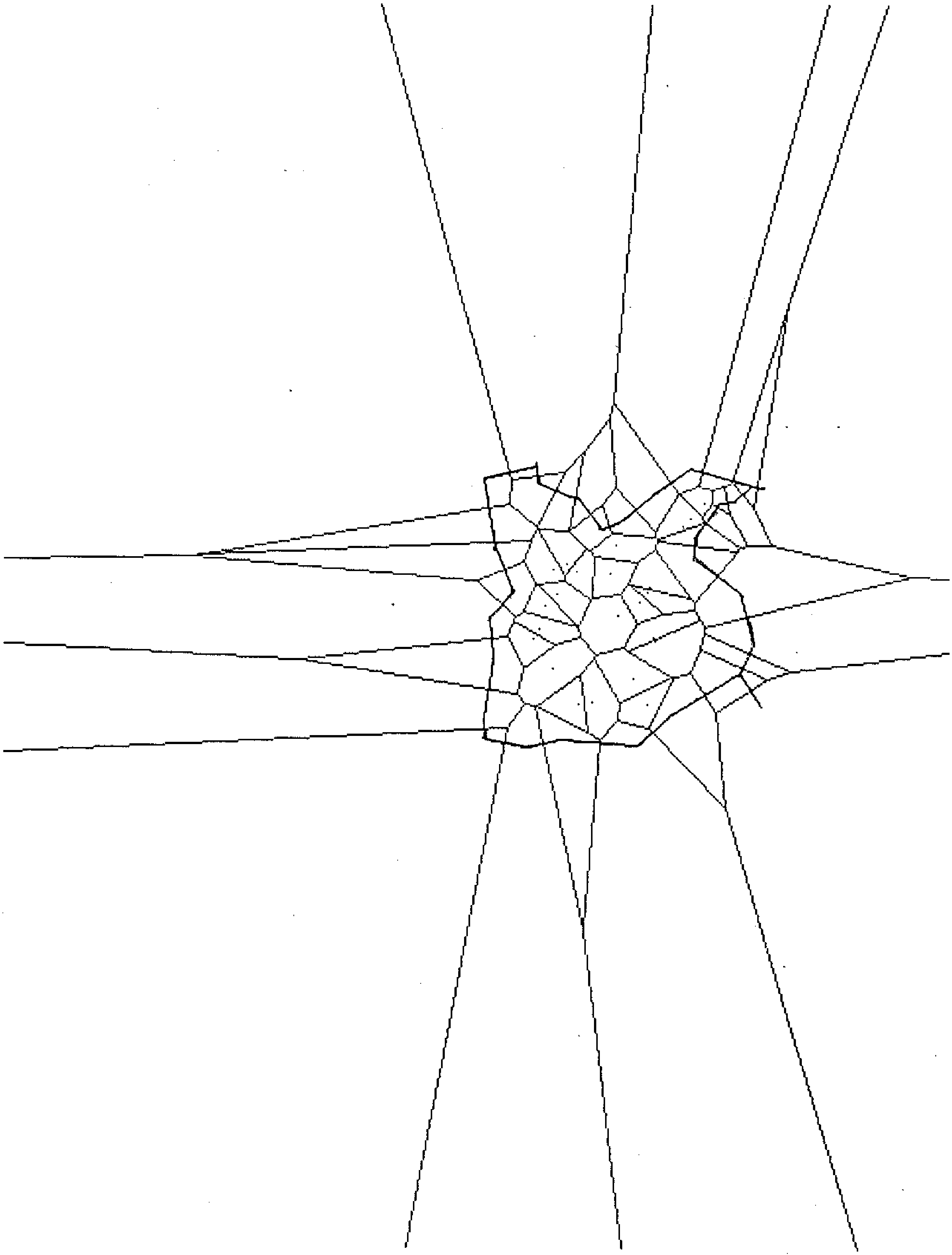


Fig. 6.3 : Area A_1 of cluster C_1

$A_1 = 0.6690$

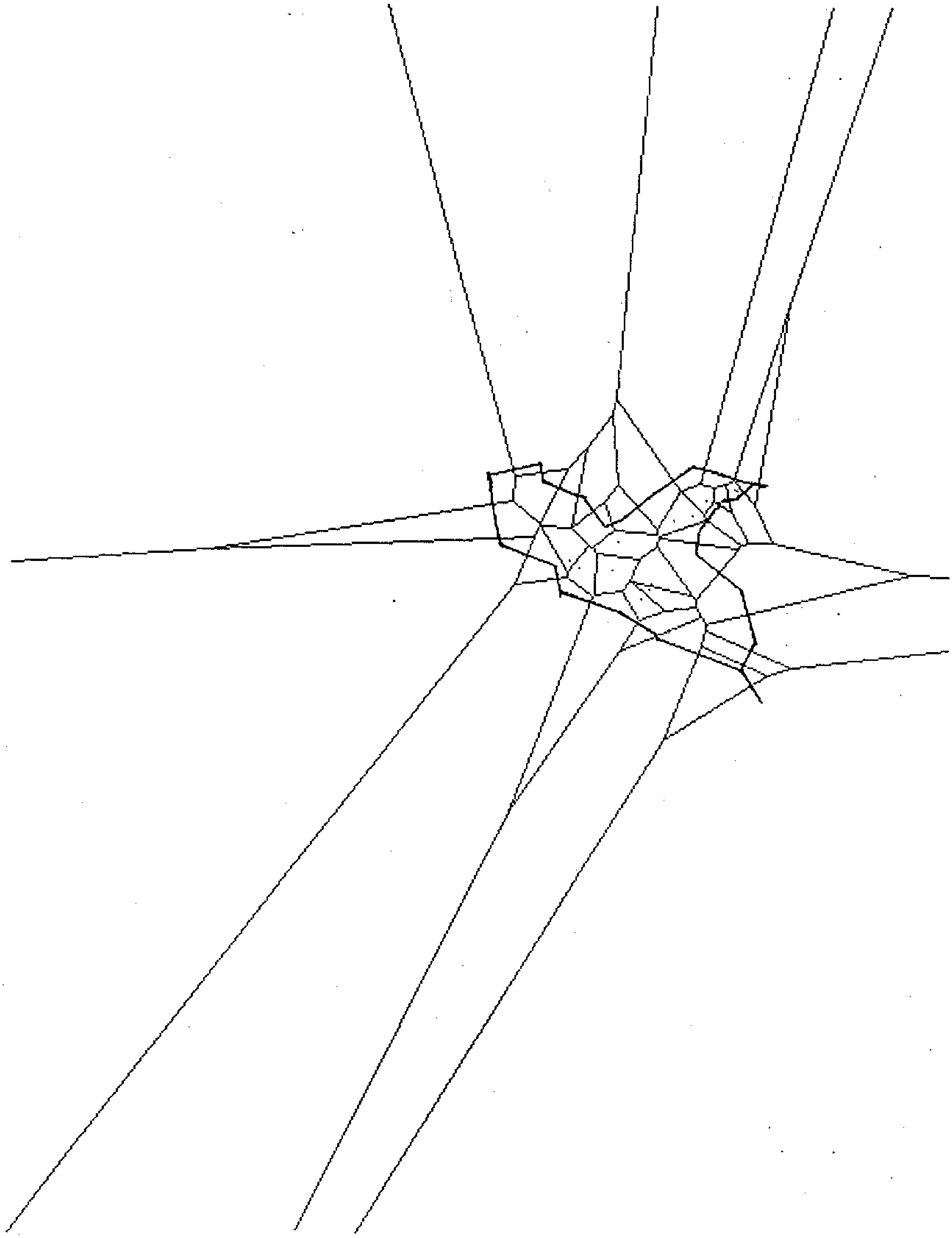


Fig. 6.4 : Area a_1 of cluster C_1

$a_1 = 0.3438$

6.1.2 Tabulation of Validity Indices

The tabulation of the Validity Indices have been divided into three Groups, each group corresponding to one of the Data-Classes defined in the last chapter. We have tabulated only the average values of the Validity Indices.

Recall that the computation of VI_1 (section 2.1) required that we know the number of extra points to be added (N-Added) to the sample so that we get the right classification. We have limited this number 'N-Added' to 25. This limit is not found to be sufficient in some cases. All we can say for such cases is that

$$VI_1 < \frac{1}{(1 + \frac{25}{n})}$$

We have marked all such entries with a '*' mark. There are cases when N-Added was found to be enough for few samples of a data set, while it was not sufficient for other samples of the same data set. All such entries have also been marked with a '*' mark. The outputs of the four clustering algorithms have been tabulated separately. It may be noted that while we have assumed that the number of clusters are known for the rest of the algorithms, the same is not true for the set-estimation method. In many cases this method gave the wrong number of clusters because of the small sample size. We have then experimented by using $\frac{h_n}{2}$, instead of h_n in our algorithm. The number of clusters obtained in both cases have been tabulated, alongwith the Validity Indices, if applicable. Otherwise such entries have been marked with '**'. Note that few entries under VI_3 are empty. For such cases VI_3 could not be computed, because more than three points were found to be co-circular, which violates the assumption on which our algorithm for the construction of Voronoi diagram is based. We have also not considered the computation of VI_3 for the data-sets under Group-II. This was because the method for the construction of Cover-Polygon is not valid when we have ring like strutures (section 4.2).

LIST OF SYMBOLS USED FOR TABULATION

- **N** : Total number of points in the sample
- **%-Mismatch** : Percentage of points which are have been misclassified
- VI_1 : Validity Index (refer Eqn. 2.1)
- VI_2 : Validity Index (refer Eqn. 2.2)
- VI_3 : Validity Index (refer Eqn. 2.3)
- $\mathbf{h_n}$:$\sqrt{\frac{I_n}{n}}$
- N - $Clus_1$: Number of clusters obtained by considering h_n
- N - $Clus_2$: Number of clusters obtained by considering $\frac{h_n}{2}$

GROUP-I

(i) $\delta = 0.8$

K-means				
n	% Mismatch	VI_1	VI_2	VI_3
50	0.0	1.0000	1.0000	1.0000
100	11.7	0.9750	0.7536	0.9681
150	1.4	0.9930	0.9867	0.9896
200	0.0	1.0000	1.0000	1.0000
250	0.0	1.0000	1.0000	1.0000

Single-linkage				
n	% Mismatch	VI_1	VI_2	VI_3
50	0.0	1.0000	1.0000	1.0000
100	0.0	1.0000	1.0000	1.0000
150	0.0	1.0000	1.0000	1.0000
200	0.0	1.0000	1.0000	1.0000
250	0.0	1.0000	1.0000	1.0000

(i) $\delta = 0.8$

Complete-linkage				
n	% Mismatch	VI_1	VI_2	VI_3
50	0.0	1.0000	1.0000	1.0000
100	0.0	1.0000	1.0000	1.0000
150	0.0	1.0000	1.0000	1.0000
200	0.0	1.0000	1.0000	1.0000
250	0.0	1.0000	1.0000	1.0000

Set-Estimation Method							
n	h_n	$NClus_1$	$NClus_2$	% Mismatch	VI_1	VI_2	VI_3
50	0.4048	2	-	0.0	1.0000	1.0000	1.0000
100	0.3250	2	-	0.0	1.0000	1.0000	1.0000
150	0.2897	2	-	0.0	1.0000	1.0000	1.0000
200	0.2678	2	-	0.0	1.0000	1.0000	1.0000
250	0.2517	2	-	0.0	1.0000	1.0000	1.0000

(ii) $\delta = 0.5$

K-means				
n	% Mismatch	VI_1	VI_2	VI_3
50	0.0	1.0000	1.0000	1.0000
100	6.0	0.9880	0.8775	0.9782
150	0.9	0.9842	0.9777	0.9813
200	0.8	0.9930	0.9800	0.9883
250	0.1	0.9975	0.9971	-

Single-linkage				
n	% Mismatch	VI_1	VI_2	VI_3
50	0.0	1.0000	1.0000	1.0000
100	0.0	1.0000	1.0000	1.0000
150	0.0	1.0000	1.0000	1.0000
200	0.0	1.0000	1.0000	1.0000
250	0.0	1.0000	1.0000	1.0000

(ii) $\delta = 0.5$

Complete-linkage				
n	% Mismatch	VI_1	VI_2	VI_3
50	0.0	1.0000	1.0000	1.0000
100	0.0	1.0000	1.0000	1.0000
150	0.0	1.0000	1.0000	1.0000
200	0.0	1.0000	1.0000	1.0000
250	0.0	1.0000	1.0000	1.0000

Set-Estimation Method							
n	h_n	$NClus_1$	$NClus_2$	% Mismatch	VI_1	VI_2	VI_3
50	0.3892	1	2	0.0	1.0000	1.0000	1.0000
100	0.3205	1	2	0.0	1.0000	1.0000	1.0000
150	0.2971	1	2	0.0	1.0000	1.0000	1.0000
200	0.2659	2	-	0.0	1.0000	1.0000	1.0000
250	0.2512	2	-	0.0	1.0000	1.0000	1.0000

(iii) $\delta = 0.2$

K-means				
n	% Mismatch	VI_1	VI_2	VI_3
50	1.2	0.9388	0.9763	0.9907
100	9.8	0.9437	0.8070	0.8891
150	4.8	0.9715	0.8710	0.9542
200	3.4	0.9810	0.9444	0.9628
250	0.0	1.0000	1.0000	1.0000

Single-linkage				
n	% Mismatch	VI_1	VI_2	VI_3
50	11.1	*	0.7307	0.8004
100	10.0	0.9377	0.7710	0.8243
150	0.0	1.0000	1.0000	1.0000
200	0.0	1.0000	1.0000	1.0000
250	0.0	1.0000	1.0000	1.0000

(iii) $\delta = 0.2$

Complete-linkage				
n	% Mismatch	VI_1	VI_2	VI_3
50	14.0	*	0.8235	0.8430
100	17.0	*	0.5643	0.9011
150	13.0	*	0.7826	0.9347
200	8.0	0.9321	0.8321	0.9431
250	0.0	1.0000	1.0000	1.0000

Set-Estimation Method							
n	h_n	$NClus_1$	$NClus_2$	% Mismatch	VI_1	VI_2	VI_3
50	0.3882	1	1	**	**	**	**
100	0.3184	1	1	**	**	**	**
150	0.2832	1	1	**	**	**	**
200	0.2619	1	1	**	**	**	**
250	0.2470	1	1	**	**	**	**

(iv) $\delta = 0.1$

K-means				
n	% Mismatch	VI_1	VI_2	VI_3
50	5.0	0.8561	0.9059	0.9384
100	15.7	0.9452	0.7100	0.7812
150	6.0	0.9930	0.9189	0.9377
200	4.9	0.9550	0.9041	0.9581
250	2.1	0.9770	0.9575	0.9879

Single-linkage				
n	% Mismatch	VI_1	VI_2	VI_3
50	43.0	*	0.0867	0.6760
100	24.5	*	0.3890	0.7321
150	45.3	*	0.0937	0.7100
200	25.0	*	0.2100	0.7621
250	10.1	*	0.2011	0.8231

$\delta = 0.1$

Complete-linkage

n	% Mismatch	VI_1	VI_2	VI_3
50	8.0	0.9230	0.8461	0.8745
100	20.0	*	0.8695	0.6783
150	45.33	*	0.0286	0.6623
200	47.2	*	0.0294	0.5731
250	15.0	0.9321	0.8023	0.8213

Set-Estimation Method

n	h_n	$NClus_1$	$NClus_2$	% Mismatch	VI_1	VI_2	VI_3
50	0.3851	1	1	**	**	**	**
100	0.3123	1	1	**	**	**	**
150	0.2845	1	1	**	**	**	**
200	0.2644	1	1	**	**	**	**
250	0.2466	1	1	**	**	**	**

GROUP-II

(i) δ-radius = 0.5

K-means			
n	% Mismatch	VI_1	VI_2
50	32.0	*	0.3240
100	34.5	*	0.3580
150	48.1	*	0.1040
200	33.5	*	0.3345
250	30.5	*	0.4041

Single-linkage			
n	% Mismatch	VI_1	VI_2
50	0.0	1.0000	1.0000
100	0.0	1.0000	1.0000
150	0.0	1.0000	1.0000
200	0.0	1.0000	1.0000
250	0.0	1.0000	1.0000

(i) δ-radius = 0.5

Complete-linkage			
n	% Mismatch	VI_1	VI_2
50	38.0	*	0.2713
100	36.2	*	0.2131
150	42.1	*	0.1865
200	41.5	*	0.3402
250	38.9	*	0.2714

Set-Estimation Method						
n	h_n	$NClus_1$	$NClus_2$	% Mismatch	VI_1	VI_2
50	0.3876	1	*	**	**	**
100	0.3077	1	*	**	**	**
150	0.2698	1	*	**	**	**
200	0.2573	2	1	0.0	1.0000	1.0000
250	0.2311	2	1	0.0	1.0000	1.0000

(ii) δ-radius = 0.4

K-means			
n	% Mismatch	VI_1	VI_2
50	36.0	*	0.2910
100	41.5	*	0.2080
150	32.4	*	0.3440
200	39.5	*	0.2445
250	32.4	*	0.3731

Single-linkage			
n	% Mismatch	VI_1	VI_2
50	43.3	*	0.2231
100	0.0	1.0000	1.0000
150	0.0	1.0000	1.0000
200	0.0	1.0000	1.0000
250	0.0	1.0000	1.0000

(ii) δ-radius = 0.4

Complete-linkage			
n	% Mismatch	VI_1	VI_2
50	36.0	*	0.2703
100	37.9	*	0.1780
150	40.1	*	0.1765
200	36.1	*	0.3302
250	38.9	*	0.2914

Set-Estimation Method						
n	h_n	$NClus_1$	$NClus_2$	% Mismatch	VI_1	VI_2
50	0.3870	1	*	**	**	**
100	0.3110	1	*	**	**	**
150	0.2750	1	*	**	**	**
200	0.2533	1	2	0.0	1.0000	1.0000
250	0.2388	1	2	0.0	1.0000	1.0000

(iii) δ-radius = 0.3

K-means			
n	% Mismatch	VI_1	VI_2
50	38.0	*	0.2677
100	42.5	*	0.1920
150	37.3	*	0.2500
200	41.0	*	0.1725
250	42.2	*	0.1581

Single-linkage			
n	% Mismatch	VI_1	VI_2
50	46.7	*	0.0934
100	28.0	0.9485	0.4642
150	0.0	1.0000	1.0000
200	0.0	1.0000	1.0000
250	0.0	1.0000	1.0000

δ-radius = 0.3

Complete-linkage			
n	% Mismatch	VI_1	VI_2
50	44.0	*	0.1523
100	37.9	*	0.2011
150	40.1	*	0.1765
200	35.1	*	0.3100
250	39.7	*	0.2814

Set-Estimation Method						
n	h_n	$NClus_1$	$NClus_2$	% Mismatch	VI_1	VI_2
50	0.4079	1	*	**	**	**
100	0.3170	1	2	28.0	0.9485	0.4642
150	0.2890	1	2	0.0	1.0000	1.0000
200	0.2643	1	2	0.0	1.0000	1.0000
250	0.2575	1	2	0.0	1.0000	1.0000

GROUP-III

(i) $\delta_1 = 0.8$, $\delta_2 = 0.5$

K-means				
n	% Mismatch	VI_1	VI_2	VI_3
50	7.3	*	0.7786	0.9471
100	2.0	*	0.9500	0.9887
150	0.46	*	0.9670	0.9894
200	2.66	*	0.9366	0.9642
250	3.3	*	0.9321	0.9833

Single-linkage				
n	% Mismatch	VI_1	VI_2	VI_3
50	5.33	0.9491	0.8672	0.9879
100	0.0	1.0000	1.0000	1.0000
150	0.0	1.0000	1.0000	1.0000
200	0.0	1.0000	1.0000	1.0000
250	0.0	1.0000	1.0000	1.0000

(i) $\delta_1 = 0.8$, $\delta_2 = 0.5$

Complete-linkage				
n	% Mismatch	VI_1	VI_2	VI_3
50	0.0	1.0000	1.0000	1.0000
100	2.1	0.9821	0.9321	0.9876
150	0.0	1.0000	1.0000	1.0000
200	0.0	1.0000	1.0000	1.0000
250	0.0	1.0000	1.0000	1.0000

Set-Estimation Method							
n	h_n	$NClus_1$	$NClus_2$	% Mismatch	VI_1	VI_2	VI_3
50	0.4581	1	*	**	**	**	**
100	0.3851	1	2	0.0	1.0000	1.0000	1.0000
150	0.3446	1	2	0.0	1.0000	1.0000	1.0000
200	0.3200	1	2	0.0	1.0000	1.0000	1.0000
250	0.3002	1	2	0.0	1.0000	1.0000	1.0000

(i) $\delta_1 = 0.5$, $\delta_2 = 0.8$

K-means				
n	% Mismatch	VI_1	VI_2	VI_3
50	34.0	*	0.2859	0.6890
100	28.0	0.9265	0.5236	0.7481
150	19.3	0.9528	0.7064	0.8732
200	13.75	0.9810	0.7891	0.8871
250	21.5	0.9826	0.6720	0.8812

Single-linkage				
n	% Mismatch	VI_1	VI_2	VI_3
50	0.0	1.000	1.0000	1.0000
100	0.0	1.0000	1.0000	1.0000
150	0.0	1.0000	1.0000	1.0000
200	0.0	1.0000	1.0000	1.0000
250	0.0	1.0000	1.0000	1.0000

(ii) $\delta_1 = 0.5$, $\delta_2 = 0.8$

Complete-linkage				
n	% Mismatch	VI_1	VI_2	VI_3
50	0.0	1.0000	1.0000	1.0000
100	0.0	1.0000	1.0000	1.0000
150	0.0	1.0000	1.0000	1.0000
200	0.0	1.0000	1.0000	1.0000
250	0.0	1.0000	1.0000	1.0000

Set-Estimation Method							
n	h_n	$NClus_1$	$NClus_2$	% Mismatch	VI_1	VI_2	VI_3
50	0.4627	1	2	0.0	1.0000	1.0000	1.0000
100	0.3757	1	2	0.0	1.0000	1.0000	1.0000
150	0.3424	1	2	0.0	1.0000	1.0000	1.0000
200	0.3160	1	2	0.0	1.0000	1.0000	1.0000
250	0.2948	1	2	0.0	1.0000	1.0000	1.0000

6.2 Conclusions

In this section we make certain observations and comparative statements about the behaviour of various algorithms for the data-sets considered by us. We also give certain properties of the cover polygon that was observed by us.

6.2.1 Validity Indices & Clustering Algorithms

The results under Group-I show that complete-linkage, single-linkage, and the set-estimation methods always give us the right classification for $\delta = 0.5$ and above. For $\delta = 0.2$ these methods give the correct classification if $n \geq 150$, while for $\delta = 0.1$ a large percentage of points are being misclassified. K-means algorithm, on the other hand showed small variation in the value of Validity-Indices when δ was increased from 0.1 to 0.8. So when the clusters are well separated, complete-linkage, single-linkage and set-estimtion methods guarantee that we always get the right classification. K-means algorithm may still misclassify some of the points, if the seed points are wrongly selected. The selection of seed points is still an open problem. However, for small values of δ the K-means algorithm showed very good results, as compared to other methods.

The results under Group-II shows that only the single-linkage, and the set-estimation methods could give the correct classification. This is to be expected since the clusters in this group are chained structures, and they are not linearly separable.

Under Group-III we have considered data-sets with three clusters, which are linearly separable. The between-class distance between clusters is more in case(ii) than in case(i). So the values of Validity-Indices are comparatively higher for case(ii) than for case(i). Single-linkage , Set-Estimation and Complete-linkage methods are found to give better results as compared to K-means for the cases considered by us.

Note that we have not tabulated the results for the triangular distribution. All algorithms were found to give the correct classification in almost all the cases considered by us.

In general, the results of Single-linkage and Complete-linkage algorithms are comparable, except for the case when the clusters are chain like structures for which case the Single-linkage is definitely better. K-means is better in case the clusters are linearly separable, but the between-class distance is comparatively small. In case the separation between the clusters is very small, the Set-Estimation gives the right number of clusters only when the number of points is very large. Otherwise, the behaviour of the Set-Estimation is same as that of the Single-linkage method.

6.2.2 Properties of Cover-Polygon

While constructing the Cover set of points from the Voronoi diagram, certain deviations were observed in the Cover-Polygon, in case the set of points are not compact. Figures 6.5(a), 6.6(a) & 6.7(a), illustrates the three possibilities.

Let us now remove the edges e_1 and e_2 in figures 6.5(a) and 6.6(a), and the vertex v_1 in fig. 6.7(a). We are left with the figures shown in 6.5(b), 6.6(b) & 6.7(b). The two polygons in fig. 6.7(b) are reconstructed from the components obtained after the removal of the vertex v_1 in fig. 6.7(a) .So in all these cases we are left with one or more closed polygons, and zero or more isolated points. The isolated points may be viewed as outliers, and the points lying within the closed polygon (including those on the boundary) form a compact structure. That is why this method gives the desired area for the computation of VI_3. So given a set of points, this method may be used to extract a set of compact structures, and isolated points. If we look at these compact structures as clusters, this might provide us with a new clustering technique. We now formally describe the above method.

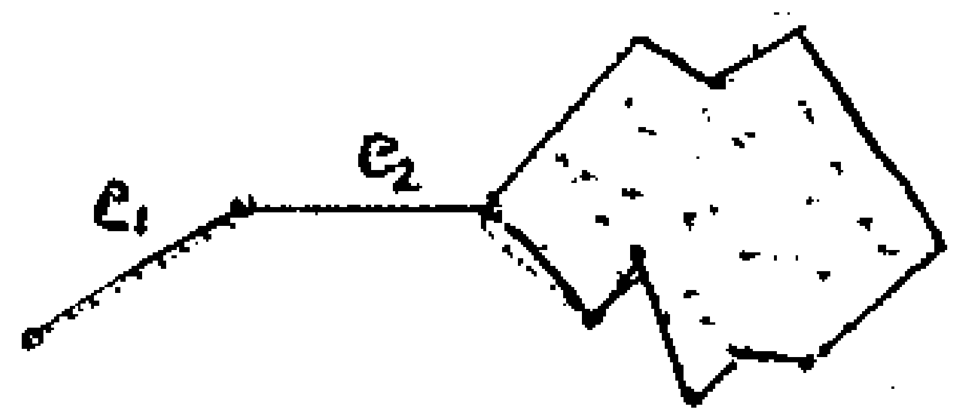


Fig. 6.5(a)

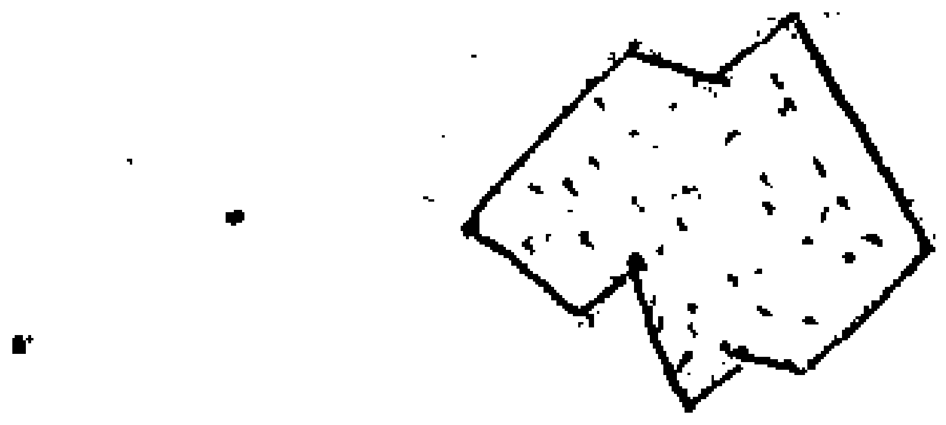


Fig. 6.5(b)

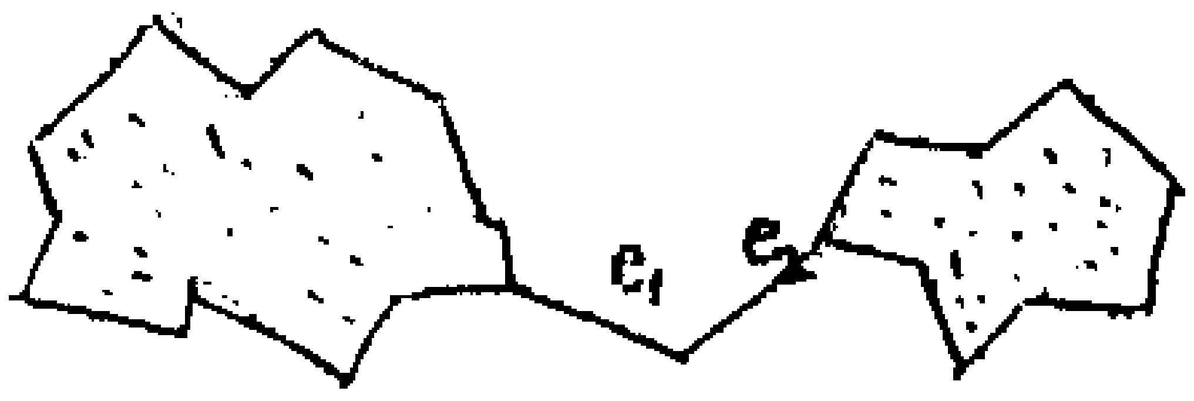


Fig. 6.6(a)

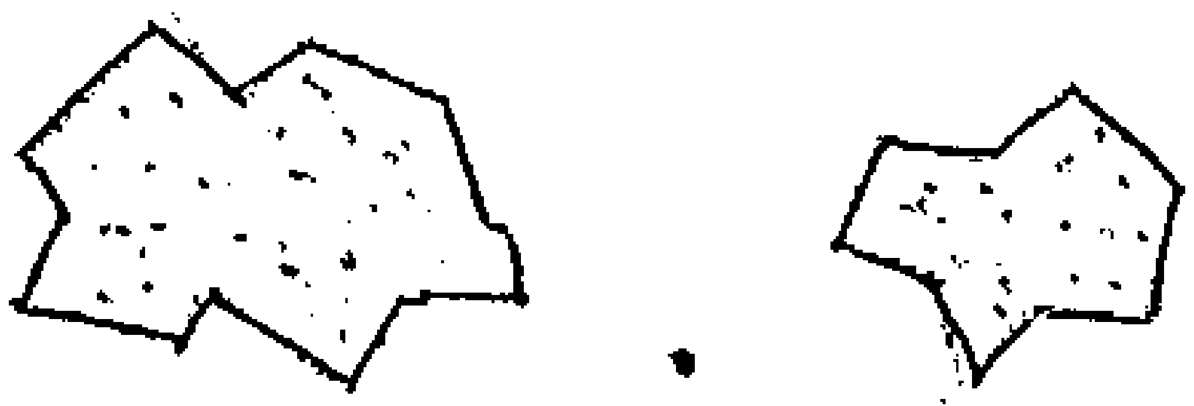


Fig. 6.6(b)

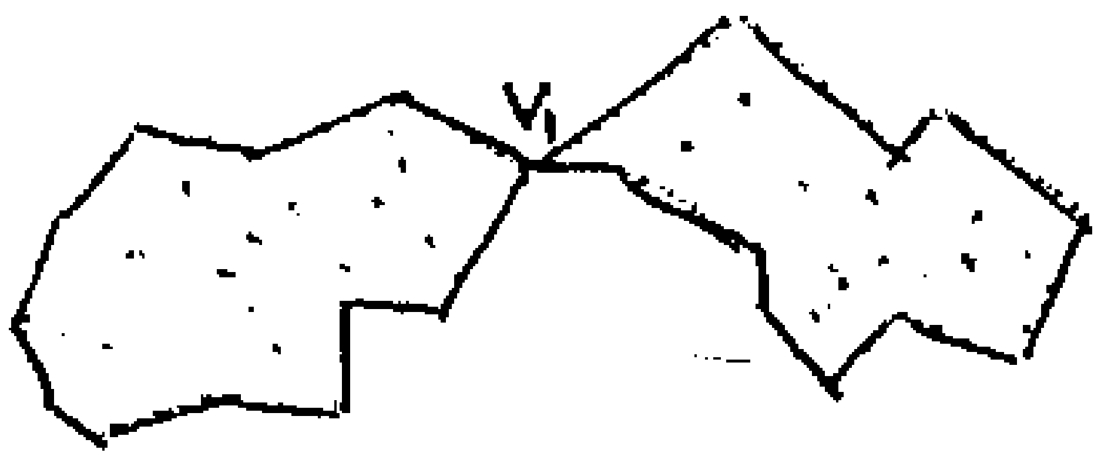


Fig. 6.7(a)

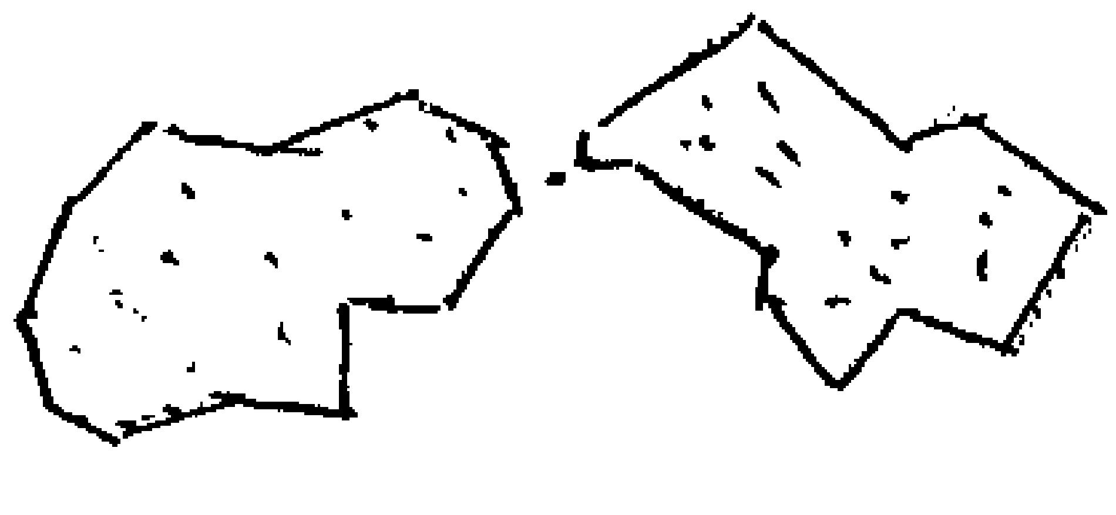


Fig. 6.7(b)

1. Construct a Voronoi-diagram for the given set of points S.

2. Construct the Cover polygon from the Voronoi-diagram, using the method described in Chapter 4.

3. Remove from the above graph, all those edges that donot form part of any closed polygon. Also remove all those vertices which form part of two or more closed polygons.

By following the above steps, we will be finally left with a set of closed polygons, and zero or more isolated points. The points within these polygons,including those on the boundaries, are the points belonging to the compact structures. These are our clusters. So we may use the above method to obtain clusters in the form of closed polygons,which simultaneously gives the shape of the clusters also. We have not carried out studies to verify the above possibility. What we can say at this stage is that this method provides us with a set of well separated and compact structures, which should give us the right classification in case the clusters are far apart and linearly separable.

6.3 Scope for further Work

The Validity Indices defined here basically provide us with some kind of measure for the misclassification of points.Since it is difficult to find the expected values, we have merely given the averages of the indices for some data sets. Further work needs to be done for other data sets, and for other clustering techniques. These indices reflect the applicability of the clustering technique for that data set and for that sample size. Note that for larger data sets the set-estimation method validates the clusters obtained by any other clustering technique. For Single-linkage and Complete-linkage techniques we assumed that the number of clusters are known apriori. Otherwise, the results would be different. The Voronoi-diagram method for finding the shape of a set of points needs further improvement, to tackle the cases where the boundary is not continuos (as in the case of a circular ring, section 4.2).

Till now we have not described how the Validity Indices that we defined can be put to actual use. We now describe the utility of the proposed Validity Indices for unknown data sets in the following steps :

1. Apply the clustering technique and obtain the clusters. Let the number of clusters be k, and the number of points in the i^{th} cluster be n_i.
2. Estimate the area of the obtained clusters by the Voronoi-diagram method.
3. Draw random samples of size n_i from i^{th} area for i= 1...k, and obtain the Validity Index.

Thus the above steps provide us with a measure of misclassification under the hypothesis that the given clustering method gives the right classification. The above method can be improved by incorporating the density estimation techniques for the clusters.

The Validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have great experience and courage.

Bibliography

[1] R.F. Ling,*An exact probability distribution on the connectivity of random graphs*, J. math. Psychol. 12, 90-98 (1975).

[2] R.F. Ling, *Probality theory of cluster analysis*, J. Am. Statist. Ass. 68, 159-164 (1973).

[3] A.K. Jain and R. Dubes, *Algorithms for Clustering Data* , Prentice Hall, New Jersey (1988).

[4] R. Dubes and A.K. Jain, *Validity Studies in clustering methodologies*, Pattern Recognition, Vol 11, pp. 235-254 (1979).

[5] M. R. Anderberg , *Cluster Analysis for Applications* , Academic Press, New York (1973).

[6] C. A. Murthy, *Consistent Estimation of Classes in $\Re^2$ in the context of Cluster Analysis*, Ph.D. Thesis, Indian Statistical Institute, Calcutta (1988).

[7] C. A. Murthy & D. Dutta Majumder, *A method of finding the boundary of a Cluster*, Data Analysis and Informatics, Vol. 4, Eds. E. Diday et al, pp. 137-148 (1986)

[8] C. A. Murthy & D. Dutta Majumder, *A method for Consistent Estimation of compact regions for cluster analysis*, Proc. of 10th International Conference on Pattern Recognition (1990).

[9] R. F. Ling & G. S. Killough, *Prabability tables for cluster analysis based on a theory of random graphs*, J. Am. Statist. Ass. 71, pp. 293-300 (1976)

[10] J. T. Tau, & R.C. Gonzalez, *Principles of Pattern Recognition*, Addison-Wesley, Reading, Massachusets (1981).

[11] F.P. Preparata & M.I. Shamos, *Computational Geometry, An Introduction*, Springer-Verlag, New York Berlin-Heidelberg Tokyo (1985).

[12] A. Rapoport and S. Fillenbaum, *An experimental study of semantic structures*, Multidimensional Scaling, A.K. Romney, R.N. Shepard & S.B. Nerlove, eds. Vol. 2, pp. 93-131. Seminar Press, New York (1972).

[13] R.J. Riddel and G.E. Uhlenbeck, *On the theory of the virial development of the equation of state of monoatomic gases*, , J.Chem. Phys. 21, pp. 2056-2064 (1953).

[14] F.J. Rohlf and D.R. Fisher, *Tests for hierarchical structure in random data sets*, Syst. Zool. 17, pp. 407-412 (1968).

[15] P.H.A. Sneath and R.R. Sokal, *Numerical Taxonomy*, W.H. Freeman, San Francisco (1973).

[16] D.W. Matula and R.R. Sokal,*Properties of Gabriel Graphs relevant to geophraphic variation research and the clustering of points in the plane*, Geographical Analysis 12, pp. 205-222 (July 1980).

[17] H. Edelsbrunner, D.G. Kirkpatrick, & R. Seidel, *On the Shape of a Set of Points in the plane*, IEEE Transactions on Information Theory, Vol. IT-29, No. 4 (July 1983).