CHAPTER 3

Quick Partition Algorithms

3.1 INTRODUCTION

Figure 3.1 consists of the outlines of 20 pieces in a child's jigsaw puzzle. The assembly of jigsaw puzzles is an instructive example of human clustering ability. A standard strategy is to first select the pieces with straight-line edges and construct the border, then to select pieces with some significant color combination and assemble these and to continue this process until most pieces are incorporated. At the end, there is usually a blah background color and these pieces are incorporated on the basis of shape. Characteristically then, some "important" variable is used as a basis for an initial crude partition and other variables are used for more detailed work within the partition.

It is difficult to formalize the patterns of color that are frequently used, but the shapes of the edges are explicitly measurable. In this puzzle (and in many), every piece has four edges and four vertices and two pieces are joined together if and only if they have an exactly matching edge. Three measurements were made for each edge:

(i) the length of the line between the two vertices,

(ii) the maximum deviation of the edge from this line into the piece, and

(iii) the maximum deviation of the edge from this line out from the piece.

There are thus twelve measurements for each piece. These measurements are given in Table 3.1.

Let $A(I, 1)$, $A(I, 2)$, and $A(I, 3)$ denote the three measurements of the first edge of the Ith piece. The first edge of the Ith piece corresponds to the first edge of the Jth piece if $A(I, 1) = A(J, 1)$, $A(I, 2) = A(J, 3)$, and $A(I, 3) = A(J, 2)$. A natural measure of distance between edges taking values $X1$, $X2$, $X3$ and $Y1$, $Y2$, $Y3$ is thus

$$D = [(X1 - Y1)^2 + (X2 - Y3)^2 + (X3 - Y2)^2]^{1/2}.$$

This is *not* euclidean distance. The distance between two pieces is the minimum distance between any two edges. If any two edges exactly coincide, the distance between the pieces will be zero.

3.2 LEADER ALGORITHM

Preliminaries. It is desired to construct a partition of a set of M cases, a division of the cases into a number of disjoint sets or clusters. It is assumed that a rule for computing the distance D between any pair of objects, and a threshold T are given. The

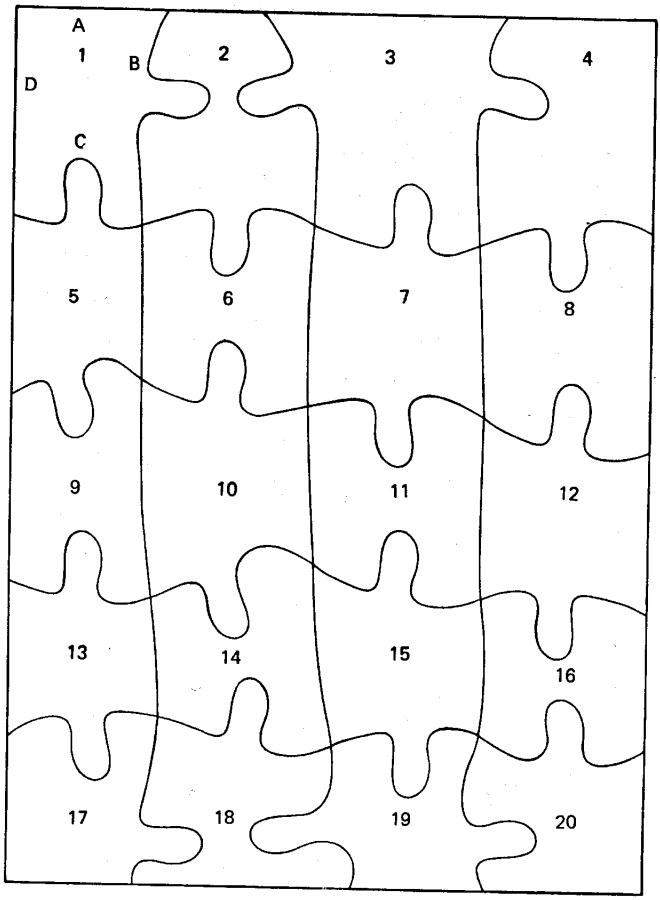

Figure 3.1 Jigsaw puzzle.

algorithm constructs a partition of the cases (a number of clusters of cases) and a leading case for each cluster, such that every case in a cluster is within a distance T of the leading case. The threshold T is thus a measure of the diameter of each cluster. The clusters are numbered 1, 2, 3, . . . , K. Case I lies in cluster $P(I)$ [$1 \leq P(I) \leq K$]. The leading case associated with cluster J is denoted by $L(J)$.

The algorithm makes one pass through the cases, assigning each case to the first cluster whose leader is close enough and making a new cluster, and a new leader, for cases that are not close to any existing leaders.

STEP 1. Begin with case $I = 1$. Let the number of clusters be $K = 1$, classify the first case into the first cluster, $P(1) = 1$, and define $L(1) = 1$ to be the leading case of the first cluster.

STEP 2. Increase I by 1. If $I > M$, stop. If $I \leq M$, begin working with the cluster $J = 1$.

STEP 3. If $D(I, J) > T$, go to Step 4. If $D(I, J) \leq T$, case I is assigned to cluster J, $P(I) = J$. Return to Step 2.

Table 3.1 Jigsaw Puzzle Measurements

Piece	Edge A			Edge B			Edge C			Edge D		
1	142	0	0	191	21	48	125	40	12	167	0	0
2	120	0	0	183	59	16	160	16	50	192	48	21
3	186	0	0	208	17	51	152	49	13	183	17	59
4	138	0	0	180	0	0	157	13	42	209	51	18
5	126	13	39	138	0	2	125	20	43	171	0	0
6	159	50	15	163	4	0	152	47	17	139	2	0
7	149	13	50	142	0	6	157	19	49	163	0	3
8	157	42	13	203	0	0	152	50	25	143	3	0
9	125	44	20	190	2	0	138	42	18	159	0	0
10	152	17	46	144	2	0	147	19	56	190	0	2
11	157	48	20	161	0	0	152	49	22	143	0	2
12	152	25	49	139	0	0	153	27	52	160	0	0
13	138	19	42	112	0	1	143	10	41	143	0	0
14	147	55	20	150	6	1	154	36	21	113	1	0
15	151	22	48	126	0	8	137	14	51	160	0	6
16	152	52	27	141	0	0	153	42	21	128	8	0
17	143	42	9	149	7	54	117	0	0	136	0	0
18	154	22	36	130	77	6	192	0	0	150	54	8
19	134	52	13	140	16	51	123	0	0	130	6	78
20	151	21	43	118	0	0	150	0	0	140	52	13

Edges are in clockwise order. There are three measurements for each edge, in hundredths of an inch (the error in each measurement is approximately $\frac{1}{100}$ in.): (i) the length of the line between the vertices, (ii) the maximum deviation of the edge from the line between vertices into the piece, and (iii) the maximum deviation of the edge from the line between vertices out from the piece.

STEP 4. Increase J to $J + 1$. If $J \leq K$, return to Step 3. If $J > K$, a new cluster is created, with K increased by 1. Set $P(I) = K$, $L(K) = I$, and return to Step 2.

3.3 LEADER ALGORITHM APPLIED TO JIGSAW PUZZLES

The algorithm requires a measure of distance, which will be the one described in Section 3.1. A threshold T is needed also. Since the error in a single measurement is approximately 1 (in hundredths of an inch), a plausible distance between two matching edges is $[2^2 + 2^2 + 2^2]^{1/2} = 3.5$. An initial threshold $T = 4$ will be used.

STEP 1. Initially $K = 1$, $P(1) = 1$, $L(1) = 1$, and $I = 1$.

STEP 2. Increase I to $I = 2$. Try the first cluster $J = 1$.

STEP 3. The distance $D(2, 1) = 1$ by matching edge B of 1 to edge D of 2. Since $D(2, 1) \leq 4$, $P(2) = 1$. Return to Step 2.

STEP 2. Increase I to $I = 3$. Try $J = 1$.

STEP 3. The distance $D(3, 1) = 19$ by matching edge D of 1 to edge A of 3. Since $D(3, 1) > 4$, go to Step 4.

STEP 4. Increase J to 2. Since $J = 2 > K = 1$, a new cluster is created with K increased by 1. Set $P(3) = 2$, $L(2) = 3$, and return to Step 2.

In this way, all cases are assigned to clusters as follows:

Cluster 1:	1	2	5	13*	16*
Cluster 2:	3	4	7		
Cluster 3:	6	10	17*		
Cluster 4:	8	11*	12	15*	
Cluster 5:	9	18*			
Cluster 6:	14				
Cluster 7:	19	20			

The pieces with asterisks do not truly have a matching edge with their cluster leader. The algorithm has identified seven true matches and five false matches. The false matches are mostly due to the algorithm rather than measurement error, because the algorithm assigns each case to the first leader to which its distance is within threshold, rather than to the closest leader. The difficulty is partly solved by reducing the threshold to $T = 2$.

Cluster 1:	1	2	5	13*	16*
Cluster 2:	3	4	7		
Cluster 3:	6	10			
Cluster 4:	8	12			
Cluster 5:	9				
Cluster 6:	11	15			
Cluster 7:	14				
Cluster 8:	17	18			
Cluster 9:	19	20			

Here there are ten true matches, and two false matches. The false matches 13 and 16 to 1 represent edges that are accidentally very close and should not be blamed on the algorithm.

It is sensible to consider clustering edges rather than pieces, since it is the similarities between edges that are used in matching pieces. The results of this clustering are given in Table 3.2. Ideally, there should be 31 clusters of pairs of matched edges, and 18 clusters of single edges corresponding to the borders of the puzzle. The computed partition contains one cluster of four edges, two clusters of three edges, 25 clusters of two edges, and 19 clusters of single edges. There are 23 true matches and 12 false matches, all of which are due to close similarities between edges without indentations. These errors seem unavoidable, whatever the algorithm.

3.4 PROPERTIES OF LEADER ALGORITHM

The positive feature of the leader algorithm is that it is very fast, requiring only one pass through the data. (It is thus not necessary to store the data in core, but it is sufficient to read it once from disk or tape.)

Several negative properties follow from the indecent haste with which objects are assigned to clusters. The first of these is that the partition is not invariant under

Table 3.2

Clusters of edges in a jigsaw puzzle using a leader algorithm with a threshold $T = 2$. False matches are denoted by an asterisk.

CLUSTER	EDGES	CLUSTER	EDGES
1	1A 13D* 16B*	26	9C 13A
2	1B 2D	27	9D 11B* 12D*
3	1C 5A	28	10B
4	1D	29	10C 14A
5	2A	30	11C 15A
6	2B 3D	31	12C 16A
7	2C 6A	32	13B 14D
8	3A	33	13C 17A
9	3B 4D	34	14B 15D
10	3C	35	14C 18A
11	4A 5B* 17D* 12B*	36	15B 16D
12	4B	37	15C
13	4C 8A	38	16C
14	5C 9A	39	17B 18D
15	5D	40	17C 20B*
16	6B 7D	41	18B 9D
17	6C 10A	42	18C
18	6D	43	19A
19	7A	44	19B
20	7B	45	19C
21	7C 11A	46	20A
22	8B	47	20C
23	8C 12A	48	20D
24	8D 11D*		
25	9B 10D		

reordering of the cases. For example, the first case is always a cluster leader. A second negative property is that the first clusters are always larger than the later ones, since they get first chance at each case as it is allocated. This property could be changed by allocating each case to the cluster whose leader it is closest to, but this change might require four or five times (or more) the distance calculations.

3.5 SORTING ALGORITHM

Preliminaries. A threshold $T(J)$ is given for the Jth variable ($1 \leq J \leq N$). Cases are partitioned into a set of clusters so that within each cluster the Jth variable has a range less than $T(J)$. The thresholds should be chosen fairly large, especially if there are many variables. The procedure is equivalent to converting each variable to a

category variable (using the thresholds to define the categories) and the clusters are then the cells of the multidimensional contingency table between all variables.

STEP 1. Order all cases I ($1 \leq I \leq M$) according to the integral part of $A(I, 1)/T(1)$.

STEP 2. Reorder all cases I ($1 \leq I \leq M$) according to the integral part of $A(I, 2)/T(2)$. Then reorder according to the integral part of $A(I, 3)/T(3), \ldots, A(I, N)/T(N)$.

STEP 3. Clusters are now the sequences of cases which have identical values of the integral parts of $A(I, J)/T(J)$ for all J.

HINT. Order by the least important variables first.

3.6 SORTING ALGORITHM APPLIED TO JIGSAW PUZZLES

This algorithm will be used for an initial sorting of edges—the matching of opposite edges would be necessary in a later step. A threshold of 10 will be used for each variable to avoid an excessively large number of clusters.

STEP 1. (Table 3.3). Order by integral part of deviation into piece divided by 10. The minimum value is 0; the maximum is 7.

Table 3.3

Inside deviation/10	Edges															
0	1A	1D	2A	3A	4A	4B	5B	5D	6B	6D	7B	7D	8B	8D	9B	9D
	10B	10D	11B	11D	12B	12D	13B	13D	14B	14D	15B	15D	16B	16D		
	17B	17C	17D	18C	19C	19D	20B	20C								
1	2C	3B	3D	4C	5A	7A	7C	10A	10C	13A	13C	15C	19B			
2	1B	5C	12A	12C	15A	18A	20A									
3	14C															
4	1C	2D	3C	6C	8A	9A	9C	11A	11C	16C	17A					
5	2B	4D	6A	8C	14A	16A	18D	19A	20D							
6																
7	18B															

STEP 2. (Table 3.4). Reorder by integral part of deviation out from piece divided by 10. The minimum value is 0; the maximum is 7.

Table 3.4

Outside deviation/10	Edges															
0	1A	1D	2A	3A	4A	4B	5B	5D	6B	6D	7B	7D	8B	8D	9B	9D
	10B	10D	11B	11D	12B	12D	13B	13D	14B	14D	15B	45D	16B	16D		
	17C	17D	18C	19C	20B	20C	17A	18D	18B							
1	1C	3C	6C	8A	9C	2B	4D	6A	19A	20D						
2	14C	2D	9A	11A	11C	16C	8C	14A	16A							
3	5A	18A														
4	4C	7C	10A	13A	13C	1B	12A	5C	15A	20A						
5	17B	2C	3B	3D	7A	10C	15C	19B	12C							
6																
7	19D															

Next reorder by the integral part of the length divided by 10 (Table 3.5). The minimum value is 11; the maximum is 20.

Table 3.5

Length/10	Edges
11	13B 14D 17C 20B
12	2A 15B 16D 19C 1C 9A 5A 5C
13	4A 5B 6D 12B 17D 18B 9C 19A 13A 15C 19D
14	1A 7B 8D 10B 11D 13D 14B 17A 20D 14A 13C 17B 7A 10C 19B
15	9D 20C 18D 3C 6C 8A 6A 14C 11A 11C 16C 8C 16A 18A 4C 7C 10A 12A 15A 20A 12C
16	1D 6B 7D 11B 12D 14B 15D 2C
17	5D
18	3A 4B 2B 3D
19	9B 10D 18C 2D 1B
20	8B 4D 3B

STEP 3. Clusters are sequences of edges with identical integer parts of $A(I, J)/T(J)$ for all J. Each sequence in a cluster is consistent with the ordering given above.

The clusters are given in Table 3.6. This clustering is not comparable to the ones based on the leader algorithm, because possible candidates for matching are not clustered together. Almost all the nonsingleton clusters are ones with straight edges, including both boundaries and internal straight edges. The other clusters must be matched to find matching edges—for example, the (15, 4, 1) cluster should be matched with the (15, 1, 4) cluster. This yields 6C–10A and 8A–4C as true matches and 3C–7C as a false match.

3.7 PROPERTIES OF SORTING ALGORITHM

The sorting algorithm is fast, requiring as many passes through the data as there are variables. The final clusters are independent of the original order of the cases (unlike clusters from the leader algorithm). They have a simple interpretation in that each variable has a range less than a given threshold $T(J)$ in every cluster.

This algorithm has a peculiar drawback in producing extremely large numbers of clusters when there are many variables. For example, with 10 variables, if every variable has an effect on the clustering (if thresholds are not set larger than the ranges of variables over all cases), there are potentially $2^{10} = 1024$ clusters. Another difficulty is that the clusters are forced to be "rectangular" by the threshold property. Thus, if a cluster is spherical, its cases will be divided among a number of rectangular cells and the cluster will not show clearly.

3.8 THINGS TO DO

3.8.1 Running the Leader Algorithm

The leader algorithm is most appropriate for very large numbers of objects for which a quick initial sorting is required. Suitable data might be the moons and planets, or life expectancies by age and sex, or the dentition of mammals. The threshold must be guessed by using the knowledge that all objects in a cluster are within threshold distance from the cluster leader. For summarizing, the cluster leader represents the cluster.

Table 3.6 Final Clusters from Sorting Algorithm

Length/10	inside deviation/10	outside deviation/10	Edges								
11	0	0	13B	14D	17C	20B					
12	0	0	2A	15B	16D	19C					
12	4	1	1C								
12	4	2	9A								
12	1	3	5A								
12	2	4	5C								
13	0	0	4A	5B	6D	12B	17D				
13	7	0	18B								
13	4	1	9C								
13	5	1	19A								
13	1	4	13A								
13	1	5	15C								
13	0	7	19D								
14	0	0	1A	7B	8D	10B	11D	13D	16B	17A	20D
14	5	2	14A								
14	1	4	13C								
14	0	5	17B								
14	1	5	7A	10C	19B						
15	0	0	9D	20C							
15	5	0	18D								
15	4	1	3C	6D	8A						
15	5	1	6A								
15	3	2	14C								
15	4	2	11A	11C	16C						
15	5	2	8C	16A							
15	2	3	18A								
15	1	4	4C	7C	10A						
15	2	4	12A	15A	20A						
15	2	5	12C								
16	0	0	1D	6B	7D	11B	12D	14B	15D		
16	1	5	2C								
17	0	0	5D								
18	0	0	3A	4B							
18	5	1	2B								
18	1	5	3D								
19	0	0	9B	10D	18C						
19	4	2	2D								
19	2	4	1B								
20	0	0	8B								
20	5	1	4D								
20	1	5	3B								

3.8.2 Improved Leader Algorithm

A one-pass leader algorithm that avoids inflated early clusters measures the distance of a new object to the leaders in reverse order and allocates it to the first leader to which it is close enough. A many-pass algorithm, which is invariant under a change of the input order and does not require the specification of thresholds, begins with an initial central object (say, the mean object on each variable). On the first pass the object furthest from this is discovered. On the second pass, objects are allocated to whichever of the first two objects they are closest, and the object furthest from its leader is discovered to be the new leader in the next pass. In this way, after K passes, K clusters will be discovered. This algorithm is analogous to the K-means algorithm, without updating of cluster leaders.

3.8.3 Number of Clusters

It is always difficult to estimate the threshold and the number of clusters to be obtained for a given threshold. A very small initial threshold will produce a large number of small clusters after expensive computation.

```
      SUBROUTINE QUICK(A,M,N,NC,THRESH,LC,LL,XMISS)
C.....20 MAY 1973
C.... QUICK SUCCESSIVELY ASSIGNS EACH ROW TO THE LAST CLUSTER FOR WHICH THE
C.... DISTANCE BETWEEN THE ROW AND THE CLUSTER LEADER IS LESS THAN THRESHOLD.
C.... IF NO CLUSTER HAS THIS PROPERTY, THE ROW BECOMES A CLUSTER LEADER.
C.... A = M BY N BORDERED ARRAY
C.... M = NUMBER OF ROWS
C.... N = NUMBER OF COLUMNS
C.... NC = NUMBER OF CLUSTERS
C.... THRESH = THRESHOLD FOR ASSIGNING AN OBJECT TO A LEADER.
C.... LL = 1 BY NC ARRAY LISTING LEADERS
C.... LC = 1 BY M ARRAY SPECIFYING LEADER FOR EACH OBJECT
C.... XMISS = VALUE TO BE TREATED AS MISSING
C.....
      DIMENSION A(M,N),LL(NC),LC(M)
      DIMENSION AA(20)
      KC=1
      LL(1)=2
      DO 20 I=2,M
      LC(I)=0
      DO 21 KK=1,KC
      K=KC-KK+1
      L=LL(K)
C.... COMPUTES DISTANCE BETWEEN ROW AND CLUSTER LEADER
      DD=0
      DC=0
      DO 22 J=2,N
      IF (A(L,J).EQ.XMISS.OR.A(I,J).EQ.XMISS) GO TO 22
      DC=DC+1
      DD=DD+(A(L,J)-A(I,J))**2
      IF(DD.GT.THRESH**2*(N-1)) GO TO 21
   22 CONTINUE
      IF(DC.NE.0) DD=(DD/DC)**0.5
      IF (DD.GT.THRESH) GO TO 21
C.... ASSIGN ROW I TO CLUSTER K IF DISTANCE BELOW THRESH
      LC(I)=K
      GO TO 20
   21 CONTINUE
      IF (KC.EQ.NC) GO TO 20
C.... CREATE NEW CLUSTER AND LEADER
      KC=KC+1
      LC(I)=KC
      LL(KC)=I
   20 CONTINUE
C.... OUTPUT CLUSTER LEADERS
      WRITE(6,4)
    4 FORMAT(16HOCLUSTER LEADERS)
      DO 60 K=1,KC
      I=LL(K)
   60 WRITE(6,3) K,(A(I,J),J=1,N)
    3 FORMAT(8H CLUSTER,I4,2X,A4,10F11.4/(18X,10F11.4))
C     OUTPUT CLUSTERS
      KC=KC+1
      DO 50 K=1,KC
      KK=K-1
      J=0
      DO 50 I=2,M
      IF (J.EQ.20) J=0
      IF (LC(I).NE.KK) GO TO 50
      J=J+1
      AA(J)=A(I,1)
   50 IF(J.EQ.20.OR.(I.EQ.M.AND.J.NE.0)) WRITE(6,1) KK,(AA(JJ),JJ=1,J)
    1 FORMAT(' CLUSTER',I5,20(1X,A5))
      RETURN
      END
```

For a metric distance [$D(I, J) \leq D(I, K) + D(J, K)$ for each I, J, K] there is a relationship between the threshold and the number of clusters. Suppose that the leader algorithm, for some order of the objects, produces K clusters at threshold T. Then, for any order of the objects, show that it produces no more than K clusters at threshold $2T$.

3.8.4 Size of Clusters

Let the data be real values with each point taken at random from [0, 1]. Then the size of the interval corresponding to the Kth cluster is smaller, in probability, than that of the $(K - 1)$th cluster; this means that for every x the Kth cluster is at least as likely as the $(K - 1)$th cluster to be less than x in size. It would be interesting to know by analysis or experiment the distribution of size of the Kth cluster. (This does not depend on the number of points, for large numbers of points.)

PROGRAMS

QUICK finds quick partition by assigning each object to the first leader object to which its distance is less than a threshold.