5. Symbolic Clustering Algorithms Using New Distance Measures

"The secret of happiness is not in doing what one likes but in liking what one has to do."
- James M. Berrie

5.1 Introduction

In Chapter 2, the conventional Agglomerative and Disaggregative clustering algorithms were proposed. Here, in this Chapter, the symbolic version of Agglomerative and Disaggregative clustering algorithms based on new distance measures will be presented. The concepts of similarity measure, dissimilarity measure, and formation of CSO which were introduced in Chapter 4 for ISODATA clustering procedure are extended in this chapter for Agglomerative and Disaggregative clustering procedures.

There exist a variety of hierarchical Agglomerative and Divisive clustering techniques. Successive merging operation is the heart of Agglomerative methods, whereas divisive methods employ successive splitting operation. Perhaps a more natural way to define clusters is by utilizing the property of mutual nearest neighbourhood between two patterns. References regarding these concepts are covered in Chapters 1 and 2. Before presenting the proposed clustering algorithms, a brief review of the existing methodologies is given in Section 5.2.

* Some parts of the material in this chapter appeared in the following research papers:


5.2 Review of Existing Methodologies


Gowda and Diday (1992) have proposed a hierarchical symbolic Agglomerative clustering algorithm. It belongs to the class of single linkage methods. The single linkage is through the identification of a mutual pair having the highest similarity values at various hierarchical levels.

Gowda and Diday (1991) present an Agglomerative clustering algorithm based on dissimilarity measure. They form composite symbolic objects using a Cartesian join operator, whenever a mutual pair of symbolic object is selected for agglomeration based on minimum dissimilarity.

Kodratoff (1988) pointed out that there should be very distinct criteria for computing similarity and dissimilarity of symbolic objects and that one should not "average" over similarity and dissimilarity. In conventional clustering of numeric objects, dissimilarity is just another aspect of similarity that leads to the view that the more similar two objects are, the less dissimilar they remain. Such a hypothesis is not true in the case of symbolic similarity and dissimilarity. It is quite possible that two symbolic objects are very much similar in some aspects and quite dissimilar in some other aspects.

Without contradicting this, Gowda and Diday (1991, 1992) have suggested a new viewpoint that in real life as the saying "Birds of the same feather flock together" goes, people come together when there are lot of similarities between them; and they go apart when they have lot of differences. They suggest that the

* Some parts of the material in this chapter appeared in the following research paper:
Similarity criterion should be used for agglomeration and dissimilarity criterion for divisive clustering. Further, Ravi (1995) has improvised on the above work of Gowda and Diday (1991) and introduced various methods of symbolic data analysis through combined measure of similarity and dissimilarity.

Motivated by the above ideas put forth by Gowda and Diday (1991, 1992), in this chapter, hierarchical agglomerative and disaggregative symbolic clustering algorithms are proposed using new similarity and dissimilarity measures.

5.3 Computational Algorithms

5.3.1 Symbolic Agglomerative Clustering Procedure

The objective of clustering is to group a set of objects into clusters such that the objects within a cluster have a high degree of similarity, while objects belonging to different clusters have a high degree of dissimilarity. Keeping this in view, an agglomerative symbolic clustering procedure based on the newly defined similarity measure for calculating mutual nearest neighborhood is presented in this section. The agglomerative symbolic clustering algorithm proceeds as follows:

1. Let \( P = \{ \{p_{11}, p_{12}, p_{13}, \ldots, p_{1X}\}, \{p_{21}, p_{22}, p_{23}, \ldots, p_{2X}\}, \ldots, \{p_{Y1}, p_{Y2}, p_{Y3}, \ldots, p_{YY}\} \} \) be a set of \( X \) by \( Y \) pixels on \( d \)-dimensions.
2. Calculate the minimum and maximum values of all the features.
3. Using the minimum and maximum values obtained in step 2, normalize the feature values of \( X \) by \( Y \) samples to number between 1 and \( k \) (user defined limit).
4. Assign the \( d \)-dimensional samples to the 2-dimensional array 'S' as described in Section 4.3.1.
5. Let \( \{X_1, X_2, \ldots, X_N\} \) be a set of \( N \) symbolic objects. Let the initial number of clusters be \( N \) having a cluster weight of 1.
6. Find the $k$-width neighbourhood pixels for each pixel by following steps from 6a to 6b.
   
   6a. Compute proximity values using similarity measure given in Section 4.3.2, between all the pixels.
   
   6b. Search the $k$-width nearest neighbors having the least proximity values.

7. Compute the MNV for every pair of pixels.

8. Determine the mutual pair having the highest similarity and form a composite object by merging the individuals of the pair (Sect. 4.3.4). Reduce the number of clusters by 1.

9. Repeat steps 6 to 8 until the $k$-width neighbourhood pixels are merged.

### 5.3.2 Symbolic Disaggregative Clustering Procedure

The Disaggregative symbolic clustering procedure starts with the entire set of samples in one class and forms the subgroups by first dividing the entire set into clusters and agglomerate the samples to these clusters based on some threshold. This procedure is continued till a stopping rule arrests further subdivisions. Strong and weak clusters will be obtained depending on the value of the threshold chosen.

The Disaggregative symbolic clustering procedure proceeds as follows:

1. Let $P = \{\{p_{11}, p_{12}, p_{13}, \ldots, p_{1X}\}, \{p_{21}, p_{22}, p_{23}, \ldots, p_{2X}\}, \ldots, \{p_{Y1}, p_{Y2}, p_{Y3}, \ldots, p_{YX}\}\}$ be a set of $X$ by $Y$ pixels on $d$-dimensions.

2. Calculate the minimum and maximum values of all the features.

3. Using the minimum and maximum values obtained in step 2, normalize the feature values of $X$ by $Y$ samples to number between 1 and $k$ (user defined limit).

4. Assign the $d$-dimensional samples to the 2-dimensional array 'S' as described in Section 4.3.1.

5. Let $\{X_1, X_2, \ldots, X_N\}$ be a set of $N$ symbolic objects. Let the initial number of clusters be 1 having a cluster weight of $N$. 
6. Find the k-width neighborhood pixels for each pixel by following steps from 6a to 6b.

6a. Compute proximity values using dissimilarity measure given in Section 4.3.2, between all the pixels.

6b. Search the k-width nearest neighbors having the least proximity values.

7. Compute the MNV for every pair of pixels.

8. Let \( T_{merge} \) and \( T_{split} (\leq 2k) \) are the thresholds of the MNV chosen for carrying out the disaggregative process. Strong and weak clusters will be obtained depending on the value of \( T_{merge} \) and \( T_{split} \).

9. Identify the pairs of pixels with MNV = \( T_{split} \) in set \( P \) having descending proximity values. Split such pair into a separate clusters \( C_i \) and \( C_j \), and increase the total number of clusters by 2.

10. Find all the mutual nearest neighbours (with MNV \( \leq T_{merge} \)) of pixel in \( C_i \) and pixel in \( C_j \). Add them to the corresponding \( C_i \) and \( C_j \) clusters and find all its mutual nearest neighbours (with MNV \( \leq T_{merge} \)). Out of them assign all those that are not already present in \( C_i \) to \( C_i \) and \( C_j \) to \( C_j \) and delete them from the set \( P \). This procedure is repeated for the third and subsequent elements in \( C_i \) and \( C_j \).

11. If all the pairs of pixels with MNV = \( T_{split} \) have been split, then make \( T_{split} = T_{split} - 1 \) and repeat steps 9 and 10 until the set \( P \) is not empty. Else repeat steps 9 and 10 until the set \( P \) is not empty.

5.4 Experimentation

Experiments have been conducted to authenticate the efficacy of the proposed symbolic clustering methodologies. Following experiments are carried out on multispectral images and the results are tabulated below. These results are compared with the results obtained from the conventional Agglomerative and Disaggregative clustering methodologies proposed in the Chapter 2 of this thesis.
The display scheme illustrated in Appendix A.3 is employed in this chapter to plot the output classification maps.

**Experiment No. 1:**

The first experiment is conducted on the Gaussian generated data set. The objects of numeric type were drawn from a mixture of normal distributions with known number of classes and classification labels so that the results validate the efficacy of the algorithm for clustering the objects and finding the number of samples in each class.

The test samples were independently generated using Gaussian vector generator. The test set is drawn from a mixture of $C$ normal distributions with mean $m_i$ and covariance matrix $COV_i$ having individual variances of 3.87 and zero covariance. The numbers of samples generated in each class and their mean values chosen are shown in Table 5.1.

The application of the proposed Agglomerative algorithm, for a neighbourhood width of $k = 5$ and the Disaggregative algorithm, for a neighbourhood width of $k = 6$ yielded six classes. As indicated in Table 5.1, there is a perfect agreement between the number of samples used for generating classes and the number of clusters predicted by the algorithms. In all the six classes, the classification results were in full agreement with the test samples generated.

**Experiment No. 2:**

The multispectral IRS (Indian Remote Sensing) satellite data covering Kundapur area of Karnataka State, India is used in this experiment for the classification purpose. Figure A.4.3 shows False Colour Composite (FCC) map of MSS image covering Kundapur area (given in Appendix A.4). This image is of size 31 X 31 with four features. The site consists of a river joining the Arabian Sea and land
area mainly made up of different vegetation.

The application of the proposed Agglomerative algorithm, for a neighbourhood width of \( k = 8 \) and the Disaggregative algorithm, for a neighbourhood width of \( k = 8 \) yielded 10 and 13 classes respectively. The classification results from the proposed Agglomerative and Disaggregative methods are given in Fig. 5.1 and Fig 5.2 respectively. The class legends, their cover types and other related information are tabulated in Table 5.2.

5.5 Critical and Comparative Analysis

It is instructive to compare the proposed symbolic clustering methods with some other methods to justify the superiority of the proposed methodologies. As a preliminary step, the proposed symbolic clustering algorithms are judiciously compared with the existing algorithms found in Takio Kurita (1991), Gowda and Diday (1991b, 1992), Gowda and Ravi (1995), and with the clustering algorithms presented in Chapter 2 of this thesis. The following paragraphs present some comparative studies made with other schemes in terms of memory and time requirements.

Takio Kurita (1991), described an efficient Agglomerative clustering algorithm using a heap strategy. In this paper, Kurita mentioned that the computational time of the Agglomerative algorithm is \( O(N^3) \) when the algorithm is implemented straightforwardly and said that it is applicable only when the number of objects is small. He introduced heap concept to speed up the searching process. The computation time to construct a heap of \( N(N - 1) / 2 \) elements, the case of clustering of \( N \) objects, is \( O(N^2 \log (N)) \). After searching the nearest pairs cluster, the algorithm requires computational time of \( O(\log(N)) \) to update or remove an element from the heap.
Gowda and Diday (1991b, 1992) proposed symbolic clustering algorithms using new distance measures. The algorithms take computational time of $[O(N^2) + O(N)]$ for clustering $N$ objects.

Gowda and Ravi (1995) presented a symbolic Agglomerative clustering based on both similarity and dissimilarity measure calculations and hence, it requires computational time of $[O(N^2) + O(N^2) + O(N)]$ for clustering $N$ objects. The clustering algorithms presented in Chapter 2 of this thesis require $O(N(w \ast h))$ and additional time of $O(Nk)$ for clustering $N$ objects (given in Sect. 2.6).

The proposed symbolic clustering algorithms required computational time of $[O(n^2) + O(n)]$ for clustering $n$ symbolic objects resulted from the data reduction phase (given in Sect. 4.3.1) and is directly proportional to the user defined limit $k$. As $n \ll N$, the computational time required by the proposed clustering algorithms is comparatively very less than the other clustering algorithms stated above.

The clustering algorithm found in Takio Kurita (1991) needs major memory requirements of about $3N^2$ and algorithm found in Gowda and Ravi (1995) requires about $2N^2$ memory space to store the proximity values. The clustering algorithms given in Chapter 2 of this thesis require about $2Nk$ memory space to store $k$-width nearest neighbours and $k$-width MNV values.

The proposed symbolic clustering algorithms required major memory space of about $2nk$ for clustering $n$ symbolic objects resulted from the data reduction phase (given in Sect. 4.3.1) and is directly proportional to the user defined limit $k$. The significance of the proposed symbolic clustering algorithms lies in their use of very little memory (i.e., $n \ll N$) as compared to the other clustering algorithms stated above. The memory requirement is the same irrespective of the dimensionality of the samples to be clustered.
Some of the limitations encountered in the earlier Agglomerative and Disaggregative clustering (Gowda, 1977, 1978b) are i) Computational time is very high in finding the $k$ nearest neighbours which are necessary for initiating merging and splitting operations, and ii) when the $k$-width neighbourhood is considered in multispectral image data, one encountered many pixels having the same proximity value with many other pixels. Hence, this leads to conflict in choosing $k$-nearest neighbour pixels, causing erroneous classification (the cause for these limitations are explained in Chapter 2). To overcome these intrinsic limitations, the proposed clustering algorithms employed data reduction technique to reduce the voluminous data and interpret these condensed data set as symbolic objects which helps in resolving the above conflicts. Beside this, strong and weak classes can be discerned by changing the neighbourhood width $k$, and the algorithms determine the number of clusters depending on this $k$-width.

Further, to authenticate the efficacy of the proposed symbolic clustering algorithms, the clusters obtained are examined for their validity using the modified Goodman-Kruskal Gamma ($\gamma$) statistic (Sect. 8.2). The level of significance values obtained using Goodman-Kruskal Gamma ($\gamma$) statistic will indicate the confidence level with which the clusters obtained can be accepted. The validation results are highlighted in Chapter 8 (Sect. 8.5).

### 5.6 Summary

Efficient nonparametric, hierarchical, symbolic Agglomerative and Disaggregative clustering procedures based on Mutual Nearest Neighborhood concept are proposed for classifying remotely sensed multispectral data. The procedure utilizes a data reduction technique to minimize the memory and computational time requirements. New symbolic non-metric distance measures and a novel method of formulation of composite symbolic objects are employed to enrich the performance of the algorithm.
Experiments are conducted on the IRS (Indian Remote Sensing) satellite data to authenticate the efficacy of the procedure. Further, to substantiate the performance of the proposed algorithms in terms of time and space, these methods are compared with conventional clustering algorithms presented in this thesis. Finally, the clustering results are validated by employing a modified Goodman-Kruskal Gamma (γ) statistic.
Figure 5.1 Classification map of MSS image near Kundapur, Karnataka, India
- Proposed symbolic Agglomerative method

Figure 5.2 Classification map of MSS image near Kundapur, Karnataka, India
- Proposed symbolic Disaggregative method
Table 5.1 Gaussian generated image details and classification results

<table>
<thead>
<tr>
<th>Class No.s</th>
<th>No. of samples generated</th>
<th>Mean feature values of generating samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>600</td>
<td>220, 150, 15</td>
</tr>
<tr>
<td>2</td>
<td>600</td>
<td>10, 50, 20</td>
</tr>
<tr>
<td>3</td>
<td>600</td>
<td>20, 215, 70</td>
</tr>
<tr>
<td>4</td>
<td>600</td>
<td>50, 220, 50</td>
</tr>
<tr>
<td>5</td>
<td>600</td>
<td>150, 115, 50</td>
</tr>
<tr>
<td>6</td>
<td>600</td>
<td>250, 40, 15</td>
</tr>
</tbody>
</table>

Table 5.1 Continued

<table>
<thead>
<tr>
<th>Class No.s</th>
<th>Symbolic Agglomerative method</th>
<th>Symbolic Disaggregative method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>600</td>
<td>600</td>
</tr>
<tr>
<td>2</td>
<td>600</td>
<td>600</td>
</tr>
<tr>
<td>3</td>
<td>600</td>
<td>600</td>
</tr>
<tr>
<td>4</td>
<td>600</td>
<td>600</td>
</tr>
<tr>
<td>5</td>
<td>600</td>
<td>600</td>
</tr>
<tr>
<td>6</td>
<td>600</td>
<td>600</td>
</tr>
</tbody>
</table>

Table 5.2 Classification results of major land covers

<table>
<thead>
<tr>
<th>Class No.s</th>
<th>Major covers</th>
<th>Symbolic Agglomerative method</th>
<th>Symbolic Disaggregative method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Water</td>
<td>544</td>
<td>374</td>
</tr>
<tr>
<td>2</td>
<td>Land</td>
<td>321</td>
<td>238</td>
</tr>
<tr>
<td>3</td>
<td>Others</td>
<td>42</td>
<td>128</td>
</tr>
</tbody>
</table>