July 11, 1990

Modification of Ordinary Clustering Algorithm to Allow for Splitting

by Michael Bellow

The ordinary clustering algorithm (CLUST) used to create cover signatures has been modified to allow for cluster splitting. In its previous form, the PEDITOR program implemented the ISODATA procedure, with an option for merging clusters whose Swain-Fu distance was sufficiently small. The user specified a maximum and minimum number of clusters. The algorithm started with the maximum number and successively merged clusters until either all remaining clusters were sufficiently separate, or the minimum number was reached. There was no means of increasing the number of clusters at any point during program execution. Furthermore, the program sometimes resulted in one or more clusters in the final output having unacceptably high variance. These high variance clusters were usually removed by the analyst later on during statistics file editing.

During the past five months, a modification that allows for cluster splitting has been devised, programmed, and tested. This modification gives the user the option of whether or not to allow splitting. A high variance cluster can be split into two "subclusters" at certain points during a program run. A cluster is only split if its variance exceeds a user-specified threshold value and the resulting two subclusters both have sufficiently smaller variance. The modification prevents high variance clusters from appearing in the statistics file created by the program, so that the analyst is spared the task of manually removing them via STATED. In addition, the effect of spurious cluster mergers occurring during CLUST execution can be undone by subsequent splits. Previously, if two clusters that should have remained separate were merged, there was no means of compensating for this at any later point in program execution.

The modified ordinary clustering algorithm proceeds as follows. The user is prompted for the same input parameters as for the old version, with several additions. The first addition is the minimum number of pixels (**m**) to retain a final class. At the end of the program run, all clusters having fewer than **m** pixels will be deleted. The user is later asked whether or not cluster splitting is to be performed. If the answer is no, then the user is prompted for the name of the output statistics file, and the old clustering procedure begins. If splitting is requested, then the program prompts the user for three parameters related to splitting, followed by the output statistics file name. The program then proceeds with the modified clustering procedure.

As with the old version, **N** initial cluster centers are defined along a diagonal of the rectangular parallelepiped in the observation space. Each pixel is then assigned to its nearest cluster center, and the iterative process of recomputing the means and reassigning pixels goes forward to convergence. Once convergence has occurred, a variance

measure is computed for each cluster containing at least **n** pixels, where **n** is a user specified number (default = **3m**, where **m** is as defined above). This measure is defined to be the largest eigenvalue of the cluster covariance matrix, which can be interpreted as the variance of the first principal component of the multivariate sample vectors forming the cluster. The program then computes, for each such cluster, the ratio of that cluster's variance to the weighted average variance (WAV) of all the clusters, where the weights are proportional to the number of pixels in the clusters. The subset of clusters for which this ratio exceeds a user-specified threshold value (default = 3) is determined; these clusters are candidates for splitting. The program selects the cluster from this subset having the highest variance and attempts to split it. No attempt is made to split clusters containing fewer than **n** pixels as such a split would likely result in one or both subclusters having too few pixels.

The procedure for splitting a cluster is based on the original ISODATA algorithm. The program treats the candidate cluster as if it were the entire data set, defining two "subcluster centers" using the parallelepiped approach. Pixels are repeatedly assigned to subclusters and the subcluster means recomputed, until convergence is achieved. The resulting two subclusters are then tested to determine whether they should replace the original cluster. If both subclusters have lower variance than the product of a user specified factor (default = 0.95) and the variance of the original cluster, and the number of pixels in each subcluster is greater than or equal to **m**, then the cluster split is declared successful. The original cluster is then replaced by the two subclusters. If either of the above conditions is not satisfied, then the cluster split is declared unsuccessful. In that case, the program returns to the previously defined subset and attempts to split the cluster having the next highest variance, using the same method. Continuing in this manner, the program successively selects clusters having lower and lower variance, until either it is able to successfully split a cluster or all clusters in the subset have failed. In the latter case, no cluster is split.

The next stage of the algorithm is the computation of Swain-Fu distances between clusters, and the decision on which, if any, cluster pairs are to be merged. This proceeds exactly as with the old version, except that a merger between two clusters just created by the latest split is not permitted. If two clusters are merged, then the iterative pixel assignment and mean computation process starts over again. Following convergence, another decision on whether to split a cluster is made, and the algorithm proceeds in this manner until either the minimum allowed number of clusters is reached or no two clusters can be merged. At that point, the program again determines the subset of clusters eligible for splitting. This time, the program attempts to split each cluster in the subset without stopping after the first successful split. Thus any number of clusters in the subset can be split. After this process is completed, all clusters containing fewer than **m** pixels are deleted, and the program terminates.

A test was performed to determine whether use of the splitting option for ordinary clustering can produce better satellite crop area

estimates. Both the old and new versions of CLUST were used to cluster the same TM and SPOT datasets. This was followed by classification using unequal priors and regression estimation. The TM, SPOT, and ground truth data were from the Iowa 1988 corn and soybean study. The results for the old version of CLUST are divided into two cases. In the first case, the statistics file created by CLUST was edited using STATED, with some clusters getting deleted due to high variance, too few pixels, or too close proximity to other clusters as measured by Swain-Fu distance. For the second case, no editing was done on the statistics file. With the new version of CLUST, the statistics file created by the program was also left alone.

The values of R^2 (regression determination coefficient) obtained for all cases are given in the table below. It can be seen that, for both TM and SPOT, the new version of CLUST gave the highest R^2 values, followed by the old version without editing. This indicates that the option to split clusters does improve efficiency.

In conclusion, the splitting option appears to be a worthwhile enhancement to the Remote Sensing Section's clustering capabilities. It eliminates the need for some of the operations associated with statistics file editing, which in the past has been highly analyst dependent. In a mathematical sense, it may provide the means for the clustering program to determine the "natural" number of groups inherent in a dataset, where previously this was not possible. Further research using additional datasets is needed in order to fully evaluate the algorithm. It should be compared with other clustering procedures such as CLASSY. In addition, the question of whether statistics file editing should be done at all, and if so to what extent, needs to be addressed.

Table 1: R^2 Values for CLUST old and new versions

Sensor	Crop	No Splitting, No Editing	No Splitting, With Editing	With Splitting, No Editing
TM	Corn	.917	.878	.928
	Soybeans	.941	.926	.943
SPOT	Corn	.773	.750	.786
	Soybeans	.850	.834	.853