Jean-Marie Beaulieu

[BeaulieuJM.ca/publi/Bea2019b]

Efficient Hierarchical Clustering for Polsar Image Analysis

Author: **Beaulieu** Jean-Marie

Conference: Advanced SAR Workshop 2019, Canadian Space Agency

Montreal (Saint-Hubert)

1-3 Oct. 2019

URL: http://www.asc-csa.gc.ca/eng/events/2019/asar-2019-workshop-on-synthetic-aperture-radar.asp

Efficient Hierarchical Clustering for Polsar Image Analysis,
Beaulieu Jean-Marie,
Advanced SAR Workshop 2019, Canadian Space Agency, Montreal (Saint-Hubert), 1-3 Oct. 2019.

Efficient Hierarchical Clustering for Polsar Image Analysis

Jean-Marie BEAULIEU
Laval University, retired professor,
301-3080 rue Antoine-du-Verdier, Quebec, QC, G1W4X9, Canada,
1-418-658-8977, jean-marie.beaulieu@ift.ulaval.ca

1. Hierarchical clustering

The agglomerative hierarchical clustering algorithm starts by assigning each data point to a distinct cluster. For N data points, we initially have N clusters. Then, iteratively, the number of clusters is reduced by 1 by merging the 2 most similar clusters. At the end, there is 1 cluster containing all the data points. At each iteration, we consider all pairs of clusters (Ci, Cj), calculate a similarity measure or distance for each pair (Di,j = D(Ci,Cj)) and merge the 2 clusters which are the most similar or have the smallest distance.

The algorithm is general but the distance D should be correctly defined for each application. D is a between cluster distance with varying cluster sizes. It could be related to inter-cluster distances d(x,y). The distance between cluster centers (mean values) is often used, D(Ci,Cj) = d(mi,mj) where mi is the mean value of the Ci cluster data points.

Iterative clustering techniques start from an initial partition and iteratively improve it. The often used K-means algorithm starts with K center positions. Each data point is then assigned to the closest center. For each center, a new position is calculated from the mean values of points assigned to the center.

2. Computing time

The computing time of the K-means is of the order of K×L×N, where K is the number of centers, L the number of iterations and N the number of pixels. The algorithm is fast because K and L are small (≤10) even if N is large. N = 10^6 for a 1000×1000 image. The computing time increases linearly with N.

The hierarchical clustering calculates a N×N distance matrix at each iteration, where N is the number of clusters. The computing time is of the order of $\sum_{n=N}^{2} n^2 = \alpha N^3$. We can expect a computing time of the order of N^2 because only the line and column associated with the last merge need to be updated. For large N, the distance matrix will be too large to be stored (10^{12} = 1 Tera).

We also need to find the minimum value of the distance matrix to identify the next cluster pair to merge. To make it $\log_2 N$ instead of N^2, with each cluster, we store the label of its best neighbor. Then, we use a **head sort tree** to find the minimum among the best neighbors. Moreover, there is no need to keep the distance matrix.

3. Polsar criterion

The log likelihood ration is used to calculate the distance between the covariance or coherency matrices of two cluster mean values.

$$D_{i,j} = 2\ln\left|(\Sigma_i + \Sigma_j)/2\right| - \ln\left|\Sigma_i\right| - \ln\left|\Sigma_j\right|$$

To reduce the computing time, a **preselection** function could be used at initialization and merging.

4. Neighbor lists and grid

For the merging steps, we can use only the k best neighbors for each cluster. The merging time is then reduced to kN. Most of the time will then be used to initialize these lists before the merging. A grid is used to reduce the initialization. A subset of the features is selected (5D) and divided into cells. For a cluster, only the neighbor cells are used to build its list.

The following table shows the computing time for different image sizes. Lists of 25 neighbors and a 5D grid are used. It is useful to use both the lists for the merging steps and the grid for the initialization of the lists. The computing time reduction is important. The hierarchical segmentation is faster, the average number of neighbors is much smaller (≈8).

	segmentation	grid+list	selection+list	selection	list	no list
300x400	0s 400ms	15s 200ms	2min 50s	8min 21s	18min 36s	42min 20s
600x400	0s 830ms	31s 500ms	11min 31s	35min 18s	74min 42s	177min 29s
600x800	1s 780ms	1min 7s 400ms	35min 3s			
1000x1000	3s 860ms	2min 31s 300ms				