Assignment 5

Hierarchical Clustering

NOVEMBER 2022

DIVYA NAIKEN (180565500)

Unsupervised Learning

Image Classification is a process where an image is categorized into a class that best represents it. The way this process is carried out is conditional on the presence of labeled training images. This is where the different approaches of Unsupervised and Supervised learning become valuable. In the absence of labels, we can use Unsupervised Learning as it is a method that uses algorithms to draw patterns on **unlabelled data**.

Unsupervised Learning algorithms such as *Clustering* can achieve the task of classifying groups. When the data collection process is complete and n-features are determined, an n-dimensional feature *space* is obtained, where each observation represents a point in this space. In the figure below, a 2-dimensional feature space is displayed where 12 observations are plotted.

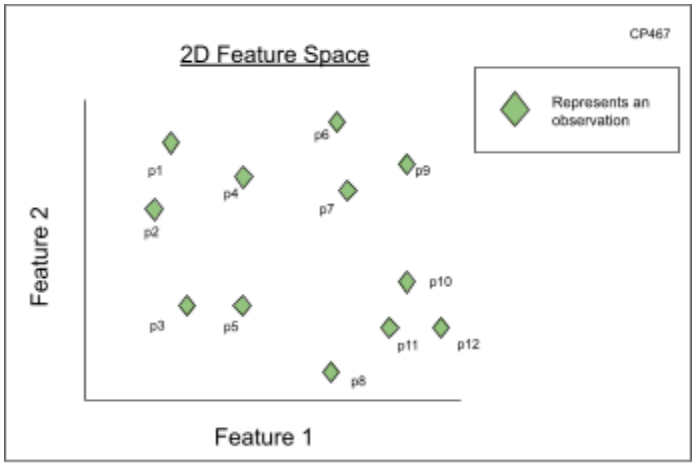

Figure 1.0 - Feature Space Plotted

At first glance, it is not obvious how to group points together in this feature space, fortunately, there are metrics that can be computed to provide more insight into the data. More specifically, the distance of the points can be calculated, and based on these distance values, certain points can be *clustered* together.

Hierarchical Clustering

In this assignment, the Agglomerative Clustering Method will be examined. Agglomerative Hierarchical Clustering is a technique where each data point is considered to be a cluster, then in each subsequent iteration, through some similarity distance measure, the nearest pair of distinct clusters are merged. This process repeats until c clusters or some threshold is obtained.

In Figure 2.0, when the minimum distance proximity measure is used, the visual breakdown of how the clusters are formed can be observed. In the beginning, each data point is assigned to 1 cluster, resulting in 12 clusters. Then, based on the distances of the points in the cluster, 4 groupings are made. The data points then continue to be grouped until 4 groups remain, as denoted by the blue rectangles.

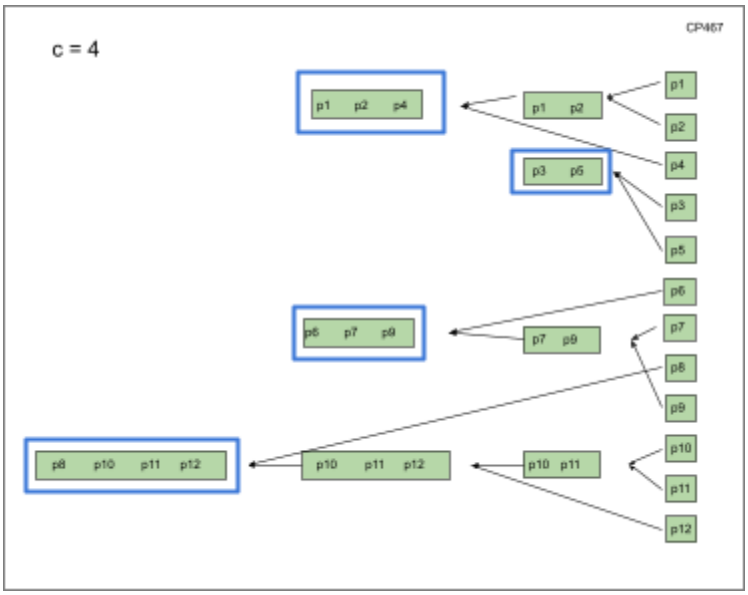

Figure 2.0 - Feature Space Clustered, c = 4

The strategy behind merging the clusters is based on whichever proximity measure best suits the problem at hand.

The similarity distance measure can be calculated in many ways, 4 methods are listed below:

$$d_{min}(X_i, X_j) = min \mid\mid x - x' \mid\mid$$

$$d_{max}(X_i, X_j) = max \mid\mid x - x' \mid\mid$$

$$d_{avg}(X_i, X_j) = \frac{1}{n_i n_j} \sum_{x_i} \sum_{x_j} \mid\mid x - x' \mid\mid$$

$$d_{mean}(X_i, X_j) = \mid\mid m_i - m_j \mid\mid$$

Source: CP467 Lecture, Agglomerative Hierarchical Clustering p.230

d_{min} is the distance between the nearest points of X_i and X_j. d_{max} is the distance between the farthest points of X_i and X_j. Figure 2.1 represents how these two measures are calculated between clusters 1 and 2.

Figure 2.1 - Computing d_{min} and d_{max}

d_{avg} is the average of all distances between pairs of points of X_i and X_j. d_{mean} is the distance between the mean of all pairs of points of X_i and X_j. Figure 2.2 and 2.2 represents how these two measures are calculated between cluster 1 and 2.

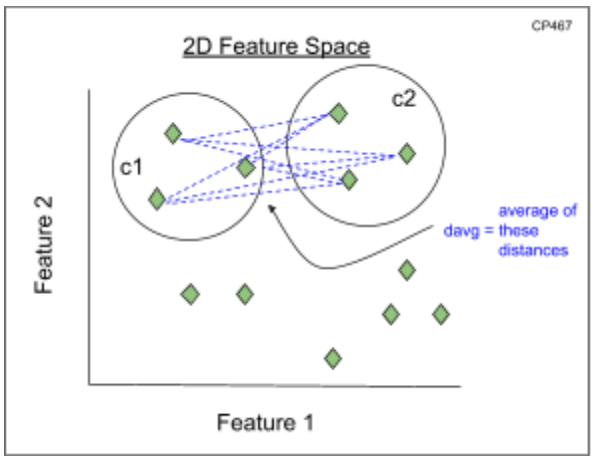

Figure 2.2 - Computing d_{avg}

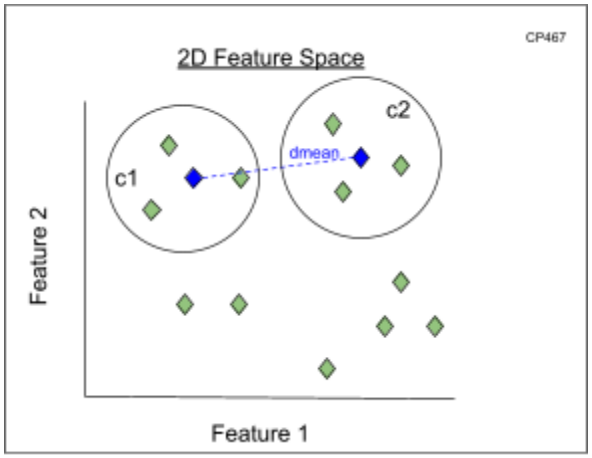

Figure 2.3 - Computing d_{mean}

Implementation

Dmin

```
def dmin(A,B):
    for coordinate1 in A:
        for coordinate2 in B:
            x = (coordinate2[0] - coordinate1[0])**2
            y = (coordinate2[1] - coordinate1[1])**2
            distance.append((x+y)**0.5)
    return(min(distance))
```

Dmax

```
def dmax(A,B):
    for coordinate1 in A:
        for coordinate2 in B:
            x = (coordinate2[0] - coordinate1[0])**2
            y = (coordinate2[1] - coordinate1[1])**2
            distance.append((x+y)**0.5)
    return(max(distance))
```

Davg

```
def davg(A,B):
   sum_distance = 0
   for coordinate1 in A:
       for coordinate2 in B:
           sum_distance += ((coordinate2[0] - coordinate1[0])**2+(coordinate2[1] -
coordinate1[1])**2)**0.5
   return(sum_distance/(len(A)*len(B)))
```

Dmean

```
#finds the minimum and maximum value in a cluster
def find_max_min(points,maxmim_val,minimum_val):
   maximum = maxmim_val
   minimum = minimum_val

   for coordinate in points:
       #max
       if coordinate[0] > maximum:
           maximum = coordinate[0]
       if coordinate[1] > maximum:
           maximum = coordinate[1]
       #min
       if coordinate[0] < minimum:
           minimum = coordinate[0]
       if coordinate[1] < minimum:
           minimum = coordinate[1]

   return(maximum,minimum)

def dmean(A,B):

   maximum, minimum = find_max_min(A, -99999, 99999)
   maximum, minimum = find_max_min(B, maximum, minimum)

   #calcualte the mean
   mean_val = (( (minimum/2)**2 + (maximum/2)**2 )**(1/2) )/ len(A)*len(B)

   return(mean_val)
```

Results

The results are summarized below:

Class 1	Class 2	dmin	dmax	davg	dmean
[(1,1), (1,2), (2,2)]	[(8,8),(9,9)]	8.49	11.31	9.91	3.02
[(1,1), (1,2)]	[(2,1),(3,1)]	1.00	2.24	1.66	1.58

Table 1.0 - Results of Distance Measures

To illustrate the point that the distance measure affects the clusters that will be merged, a new cluster named, cluster C = {(4,6),(5,8)} will be added to the space containing Class 1 and Class 2 from the first row in Table 1.0.

Class 1	Class 2	dmin	dmax	davg	dmean
A=[(1,1), (1,2), (2,2)]	B=[(8,8),(9,9)]	8.49	11.31	9.91	3.02
A=[(1,1), (1,2), (2,2)]	C=[(4,6),(5,8)]	4.47	8.06	6.21	2.69
B= [(8,8),(9,9)]	C= [(4,6),(5,8)]	3.00	5.83	4.36	4.92

Table 1.1 - Distance Measures for Clusters A, B and C

It is now evident that depending on the distance measure, the merging will differ. If the distance measure is dmin, the clusters B and C will be merged, however if the distance measure is dmean, the clusters A and C will be merged.

Remarks

In Computer Vision, there are many applications for using Unsupervised Learning. It is important to note that there are tasks where it is an obligation to use Unsupervised Learning, but there are also tasks that benefit from a resource point of view to deploy these algorithms. For instance, when there are many records of data, the process of labeling or categorizing data can be very costly when done accurately. As a result, automating the process of categorizing data may be beneficial. As discussed, Agglomerative Clustering is one of the ways to do this.

In this assignment, only 4 distance measures were discussed, but other metrics exist. There is a commonly used technique named Ward's method which minimizes the increase in variance around the centroids when 2 clusters are combined. There are advantages to using this method, but, ultimately, choosing the best one is based on the application. Some methods work best for particular types of data (binary, continuous etc…), but there is not a one-size-fits-all answer - experimentation is the best way to arrive at a conclusion.