DATA MINING ASSIGNMENT 10
(Image Clustering)

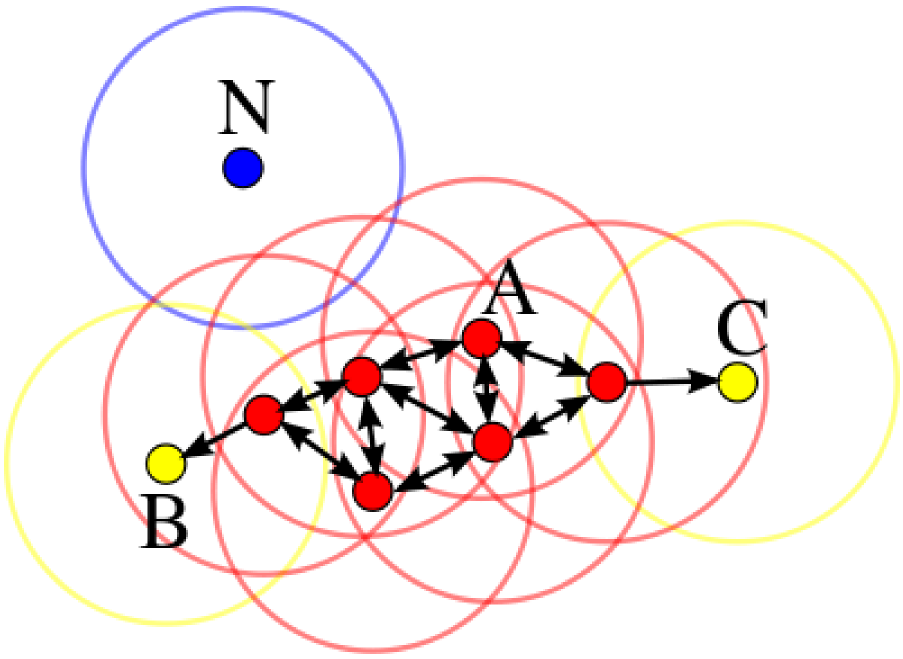
Debarati Das

1PI13CS052

OVERVIEW (PART A)

Density based clustering methodologies are preferred in case of images which are arbitrary shaped and where the noise lies within the cluster itself(“convex regions”).

DBSCAN is a density-based clustering algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away).



In this diagram, minPts = 3. Point A and the other red points are core points, because at least three points surround it in an ε radius. Because they are all reachable from one another, they form a single cluster. Points B and C are not core points, but are reachable from A (via other core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor density-reachable.

Advantages

1. DBSCAN does not require one to specify the number of clusters in the data a priori, as opposed to k-means.
2. DBSCAN can find arbitrarily shaped clusters.
3. DBSCAN has a notion of noise, and is robust to outliers.
4. DBSCAN requires just two parameters and is mostly insensitive to the ordering of the points in the database.

Disadvantages

1. DBSCAN is not entirely deterministic: border points that are reachable from more than one cluster can be part of either cluster, depending on the order the data is processed.
2. The quality of DBSCAN depends on the distance measure used in the function regionQuery(P, ε). The most common distance metric used is Euclidean distance.
3. DBSCAN cannot cluster data sets well with large differences in densities, since the minPts-ε combination cannot then be chosen appropriately for all clusters.
4. If the data and scale are not well understood, choosing a meaningful distance threshold ε can be difficult.

ANSWERS TO QUESTIONS

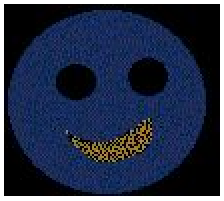
1. **What criterion did you use to decide whether a pixel is a core object?**

   A point *p* is a core point if at least minPts points are within distance ε of it, and those points are said to be directly reachable from *p*. No points are *directly reachable* from a non-core point.
   Therefore, Pixel can be decided as a core object based on if it's ε -neighbourhood contains at least "minPoints" objects.

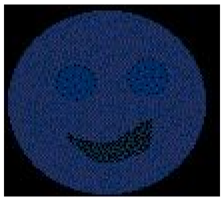
2. **What values of epsilon (in pixel space and in color space) and MinPts were used and why?(Note that you may want to try out several sets of values before deciding on one set)**

   As the values were normalised, after many runs of the program, the optimal values selected were:
   - **Epsilon : 0.3**
   - **minPoints : 15**

   As you increase epsilon value the number of points you search for increases by e^2 in the space around the current point, so it takes longer to search for the points. When epsilon was 0.2 and minpoints=10 : The picture seemed more well defined, as shown below , but the noise was increasing to 4003 while clusters were found to be 10.



   On increasing epsilon value to 0.4 , number of clusters was 1 , which defeated the purpose of clustering. Now further experimentation and tweaking of epsilon and minPoints yielded a noise level of 3460 and 4 clusters. As from mere observation we can see that there are 4 distinct shapes in the picture, we can conclude this is the optimal setting.



3. **What logic was used to determine whether a pixel was in the neighborhood of another pixel?**

   This is done by observing the color and pixel distance of one pixel to another. The distance involves using euclidean distance in both cases to calculate the value.