Homework 6:

INFO 523

Group 1

Matt Miller

Kai Blumberg

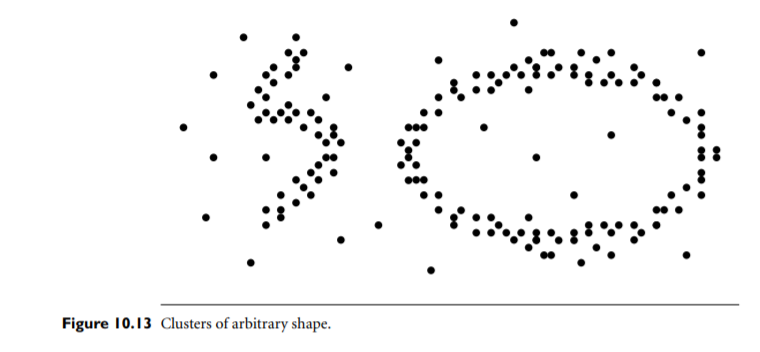
**10.4**: For the k-means algorithm, it is interesting to note that by choosing the initial cluster centers carefully, we may be able to not only speed up the algorithm’s convergence, but also guarantee the quality of the final clustering. The **k-means**++ algorithm is a variant of k-means, which chooses the initial centers as follows. First, it selects one center uniformly at random from the objects in the data set. Iteratively, for each object p other than the chosen center, it chooses an object as the new center. This object is chosen at random with probability proportional to dist(**p**)2, where dist(**p**) is the distance from **p** to the closest center that has already been chosen. The iteration continues until k centers are selected.

Explain why this method will not only speed up the convergence of the k-means algorithm, but also guarantee the quality of the final clustering results.

Consider the optimal cluster centers that are all equidistant from one and cluster centers paired up in close proximity. For the initial assignment, the close together points will lose further away points that the optimal clusters acquired. After updating means, the new centers for the close points will be a greater distance away than the new centers for the optimal points. As points are reassigned, the centers will slowly move apart and require reassignment while optimal centers will require fewer cycles to reach stopping criteria. There’s a chance that

**10.12**: Present conditions under which density-based clustering is more suitable than partitioning-based clustering and hierarchical clustering. Give application examples to support your argument.

Density-based clustering is more effective when the clusters are arbitrarily-shaped rather than spherical. While partitioning-based and hierarchical clustering methods use the distance between points to form clusters, which only allows the identification of spherical clusters, density-based clustering forms clusters based on the density of points. High-density regions are separated by low-density regions, and the high-density regions are the clusters. Below is a picture from the book showing different shaped clusters. Density-based methods work well for applications that generate lots of noise, as noise is generally low density and can be weeded out.



**10.16**: Describe each of the following clustering algorithms in terms of the following criteria: (1) shapes of clusters that can be determined; (2) input parameters that must be specified; and (3) limitations.

**(a) k-means**

1. A: spherical-shaped clusters
2. B: k, the number of clusters
3. C: cannot correct erroneous merges or splits, initial selection of cluster seeds can bias the process, outliers can significantly affect clusters, may stop at local optimum instead of global optimum, only works well for smaller datasets

**(b) k-medoids**

1. spherical-shaped clusters
2. k, the number of clusters
3. not scalable to large datasets unless alternative algorithms are used (I’m assuming we’re talking about PAM here)

**(c) CLARA**

1. spherical-shaped clusters

(2) k, the number of clusters

(3) Selection of samples can bias the clustering and may miss one of the optimal medoids of the entire dataset

**(d) BIRCH**

1. Spherical-shaped clusters
2. Branching factor and threshold
3. Each node in the CF-tree can hold a limited number of entries, so the CF-tree may not correspond to what a user views as a good or natural cluster

**(e) CHAMELEON**

1. Arbitrarily-shaped clusters

(2) Just the n-samples, d-dimensions input data

(3) Time complexity of O(n^2) can be slow for large datasets

**(f) DBSCAN**

1. Arbitrarily-shaped clusters
2. Epsilon, the maximum reachable distance for a point to be a part of a cluster, and MinPts, minimum number of points in a neighbor to be considered a cluster
3. Time complexity of O(n^2) in the worst case can be slow for large datasets