CMSC 691 – Intro. to Data Sci.
HW – 5

Akshay Peshave (peshave1@umbc.edu)

10.2:

(a) Cluster centroids after first iteration are:
(2,10), (6,6), (1.5,3.5)

(b) Final three clusters are :
Cluster 1 – {A1, B1, C2}
Cluster 2 – {A3, B2, B3}
Cluster 3 – {A2, C1}

10.10:

BIRCH[1,2] uses cluster radius or diameter as a threshold to control the boundaries of each cluster. These thresholds lead to clusters spherical in shaped only and may not reflect actual (logical/intuitive) clusters contained in the data.

OPTICS[1] finds a cluster ordering for data points. Cluster membership is computed based on minimum reachability distances of data points from core points. Reachability distances in density based clustering are not constrained by a euclidean radius threshold. They are based on transitive paths to the centroid or core points of the cluster constrained by a parameter which is a function of the acceptable cluster density. Thus, resulting clusters may be of arbitrary shapes.

Modifying BIRCH for arbitrary cluster shapes:
The leaves in a BIRCH cluster tree represent clusters at the finest granularity in the cluster hierarchy. The euclidean diameter constraint is applied at the leaves of the cluster tree, thus restricting the clusters to spherical shapes.

We can modify BIRCH by modifying the criteria for adding data points to leaf nodes. We define a parameter **ε** which is proportional to the density threshold for clustering. It should represent, in euclidean space, the tolerance of a data point's distance from the boundary of an existing cluster or from it's centroid. We can compute a metric RC

$$RC_i = \underset{leaf}{argmin}\left(\sqrt{(x_i - x_{0,leaf})^2}, \frac{D_{leaf}}{2} - \sqrt{(x_i - x_{0,leaf})^2}\right)$$

for each new data point that needs to be inserted into the cluster tree. If $RC_i < \varepsilon$ then add the data point to the corresponding leaf node, else create a new leaf node with the data point in it. We do not split leaf nodes based on a radius or diameter threshold, thus avoiding fitting only spherical clusters. Smaller values of **ε** will lead to tighter, denser and potentially more number of clusters.

In the second phase of BIRCH, an agglomerative clustering method of choice should be configured to merge dense clusters whose boundaries lie within **ε** proximity while not discarding sparsely populated clusters as outliers. The resultant clusters will be closer in density terms to those created by density-based clustering methods while not necessarily being spherical in shape.

Although not completely accurate for all data distributions, this method has the potential to yield non-spherical shaped clusters while using the original BIRCH clustering features and have the same computational complexity.

10.12:

Partition-based and hierarchical clustering methods use euclidean distance neighborhood thresholds to define cluster membership which result in clusters which are constrained radius neighborhoods or multi-dimensional spheres. The threshold is an input parameter in addition to others such as number of clusters, minimum cluster membership or hierarchy dimensions. These parameters may need to be estimated for each dataset to get optimal values which represent actual cluster structure.
Additionally, euclidean distance criteria limit the effectiveness of these approaches to data points which separable in n-dimensional space (where n is the dimensionality of the data) by n-dimensional planes or constrained radius neighborhoods. Thus, a set of data points surrounding another set (see Fig 10.12.a) or two sets of points intertwined with each other (see Fig 10.12.b) are not separable.

Density-based clustering methods are best suited for cases where number of clusters are needed to be defined based on the data and/or the data distribution is as described above. These methods cluster points based on “reachability” from core points, where core points represent the localized continuity of a dense cluster. The reachability threshold parameter constrains the sparseness of the clusters and is the only parameter that needs to be estimated and specified for the given data. This metric ensures separation of data points in the two aforementioned cases and does not necessarily restrict the cluster boundaries by a predefined radius.

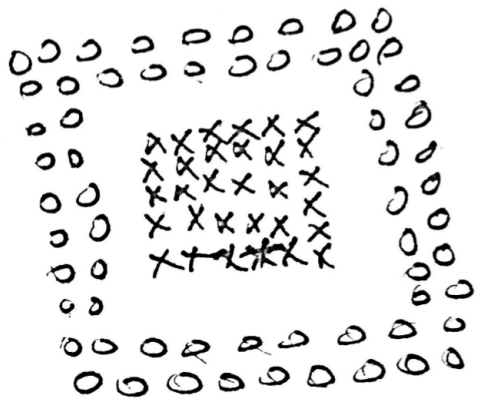

Fig. 10.12.a

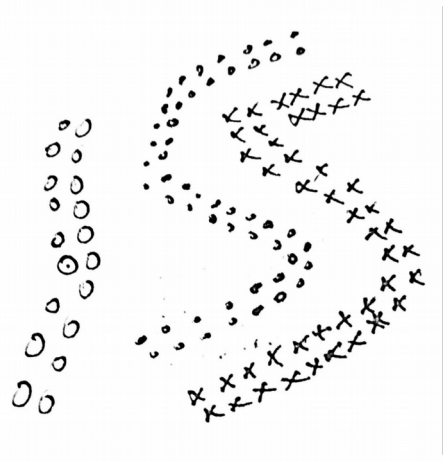

Fig. 10.12.b

Consider the use-case of applying clustering methods to detect contours or objects, based on edges or shading, in a hand-drawn sketch. Density-based clustering methods are a better candidate than partition or hierarchical clustering. Such a sketch will potentially exhibit points distributed as depicted in the figures above and density-based clustering techniques should be able to identify the objects or continuous contours better.

10.18:

An obstacle prevents two households from being placed in the same cluster. As input to the clustering algorithm we should provide a list of all pairs that are separated by obstacles. In every iteration of k-means clustering this list should be checked for the new household being considered for cluster membership and all households already part of the cluster. We can store all obstructed pairs in a hash-map which ensures single pair lookup time is $O(1)$, with the net lookup time being of the order O(no. of points clustered) in the worst case. Since, number of points clustered follow the progression $1+2+3+\ldots.+(n-1)=\frac{1}{2}n(n-1)$, the additional time overhead in the worst case is of the order $O(n^2)$. The space complexity increases by an order of $O(p)$, where p is the number of obstructed household pairs. This approach is the same as introducing a "Cannot-link" constraint discussed in [3]

The minimum cluster size constraint can be addressed adding an extra step at the end of the regular k-means approach as described below. We can initialize cluster centroids in two ways:

1. We initialize k-means with $k=\left\lceil \frac{\#\text{ households}}{\text{min\_cluster\_size}} \right\rceil$, where the k initial centroids are chosen using the obstructed household pairs list. The list can be used to select k/2 pairs at random and use these points as the initial centroids.
2. We initialize k-means with $k=\text{max\_allowable\_ATMs}$ with centroids initialized using the obstructed household pairs list as above. Then, we let k-means complete the clustering process.

In both cases, if we encounter a household that cannot be assigned to preexisting clusters in any phase of the refined k-means algorithm, we assign it as a centroid for a new cluster and proceed. We use the resultant clusters to identify clusters which do not meet the minimum size constraint. For each such cluster, we proceed by assigning each household therein to one of the other clusters while recomputing centroids. Additionally, we attempt to merge the smallest clusters with other clusters in the last iteration to minimize the number of clusters. This refinement adds to the time complexity of k-means. Additional time to the order of $O(n^2)$ per iteration is incurred.

REFERENCES:

[1] Jiawei Han. 2005. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[2] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. In Proceedings of the 1996 ACM SIGMOD international conference on Management of data (SIGMOD '96), Jennifer Widom (Ed.). ACM, New York, NY, USA, 103-114. DOI=http://dx.doi.org/10.1145/233269.233324

[3] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. 2001. Constrained K-means Clustering with Background Knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning* (ICML '01), Carla E. Brodley and Andrea Pohoreckyj Danyluk (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 577-584.