Clustering

Lecture-9

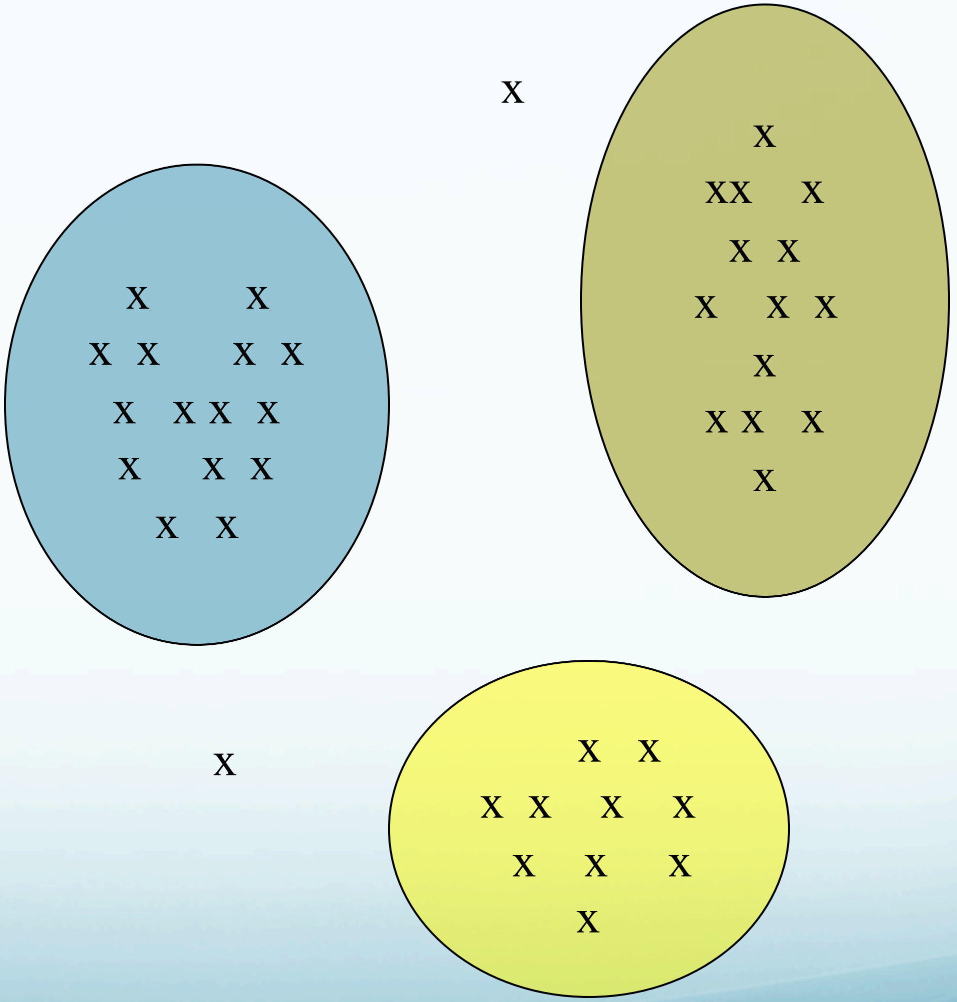
Barna Saha

Acknowledgement: Some of the slides taken from
Jeff Ullman's course on Mining Massive Datasets

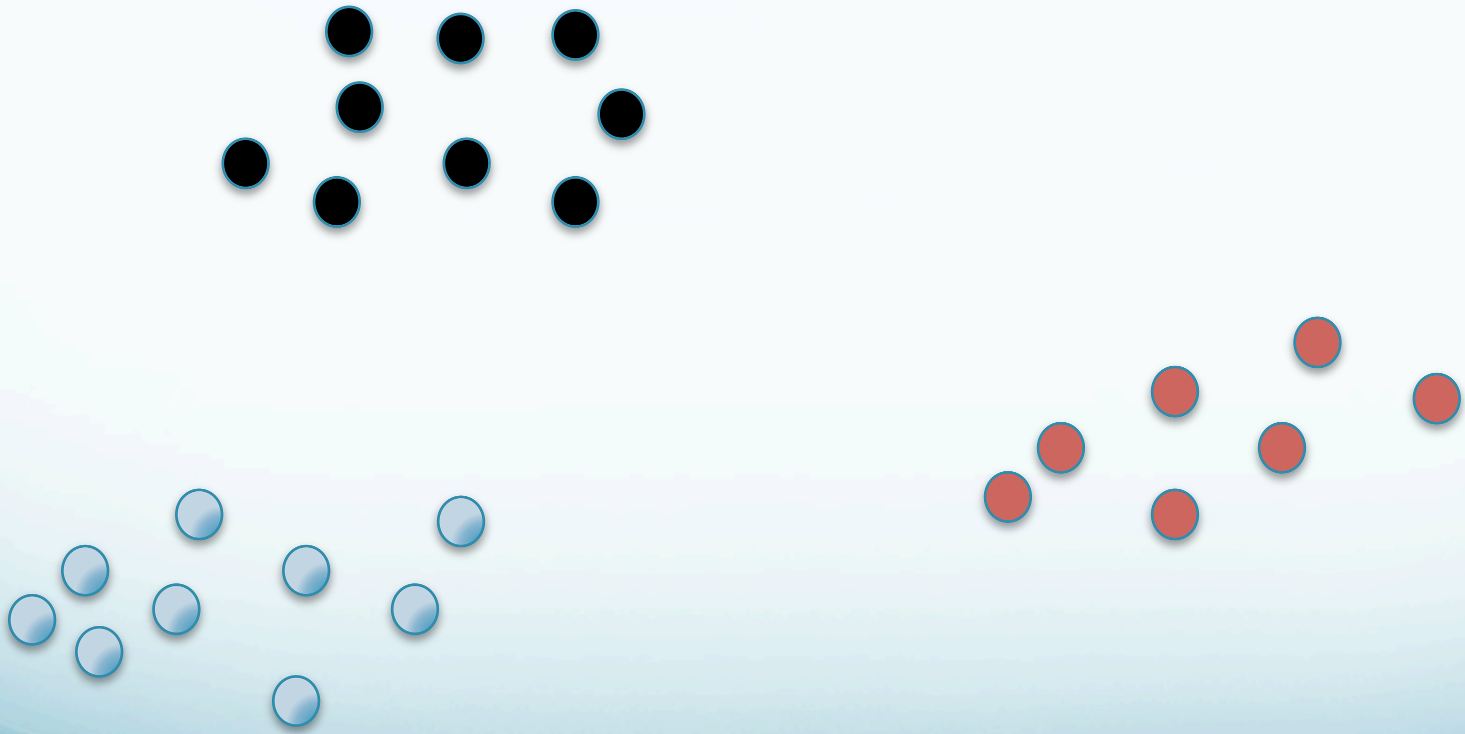
The Problem of Clustering

- Given a set of points, with a notion of distance between points, group the points into some number of *clusters*, so that members of a cluster are "close" to each other, while members of different clusters are "far."

Example: Clusters



Clustering in Low Dimensional Euclidean Space is Easy



Modern Clustering Problem

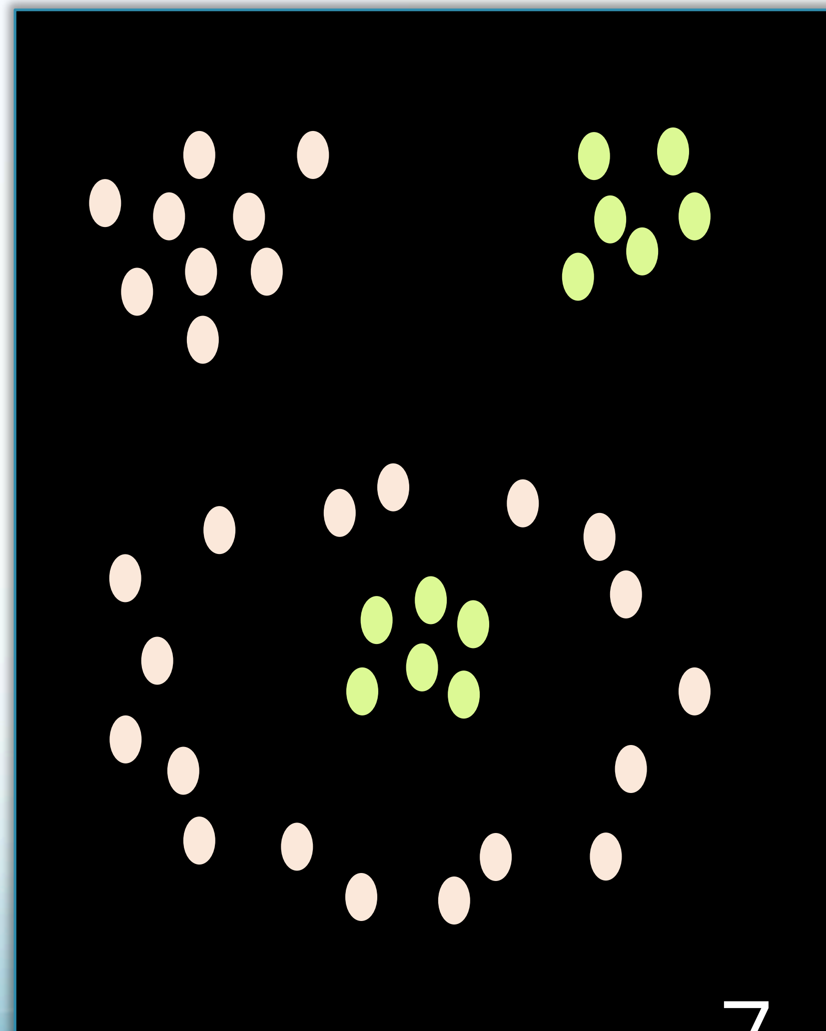
- May involve Euclidean spaces of very high dimension.
- Non Euclidean space: Jaccard distance, Cosine Distance, Hamming Distance, Edit Distance etc.
- Example:
  - Cluster documents by topics based on occurrences of unusual words
  - Cluster moviegoers by the type or types of movies they like
  - Cluster genes by their sequence similarity

Clustering Strategies

- Two fundamentally different approaches
  - Hierarchical or Agglomerative Clustering
    - Start with each point in its own cluster
    - Merge clusters based on ``closeness''
  - Point Assignment
    - Start with some clusters (possibly empty)
    - Consider points and insert them in appropriate clusters

Which is Better?

- Point assignment good when clusters are nice, convex shapes.
- Hierarchical can win when shapes are weird.



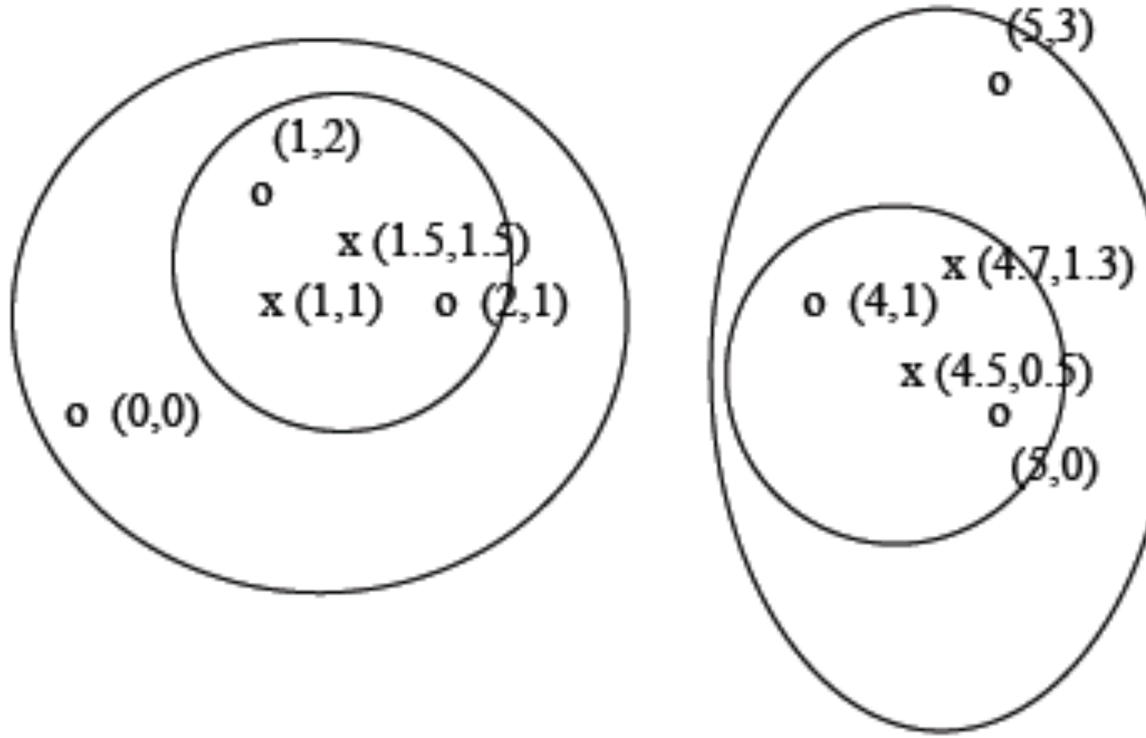
Hierarchical Clustering

- Two important questions:
  1. How do you determine the "nearness" of clusters?
  2. How do you represent a cluster of more than one point?

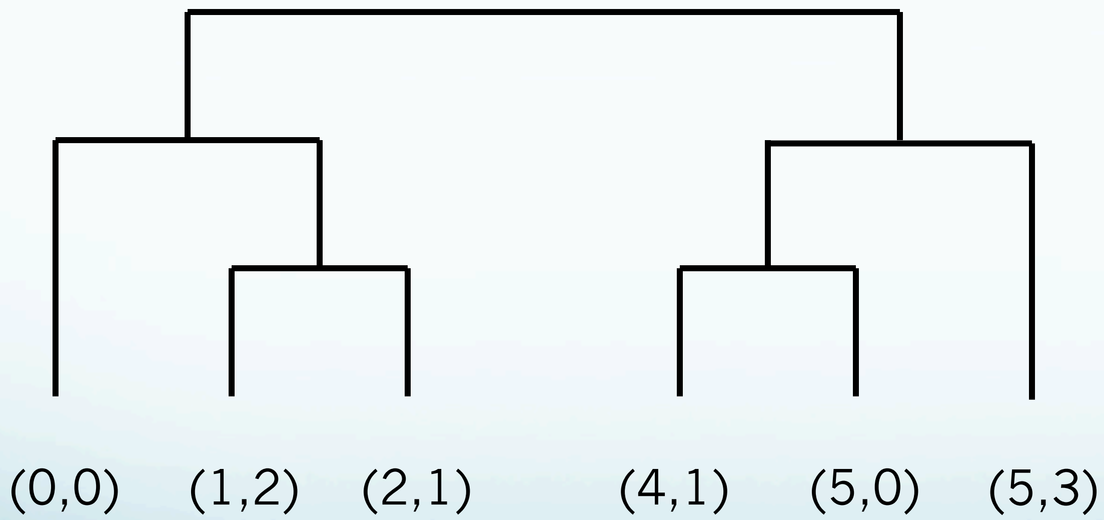
Hierarchical Clustering – (2)

- Key problem: as you build clusters, how do you represent the location of each cluster, to tell which pair of clusters is closest?
- Euclidean case: each cluster has a *centroid* = average of its points.
  - Measure intercluster distances by distances of centroids.

Example



Tree showing the grouping of points



And in the Non-Euclidean Case?

- The only “locations” we can talk about are the points themselves.
  - I.e., there is no “average” of two points.
- Approach 1: *clustroid* = point “closest” to other points.
  - Treat clustroid as if it were centroid, when computing intercluster distances.

“Closest” Point?

- Possible meanings:
  1. Smallest maximum distance to the other points.
  2. Smallest average distance to other points.
  3. Smallest sum of squares of distances to other points.
  4. Etc., etc.

Efficiency of Hierarchical Clustering

- Start by computing $O(n^2)$ distances
- Subsequent steps taken $O((n-1)^2)$, $O((n-2)^2)$,....
- Total=$O(n^3)$
- Can be reduced to $O(n^2 \log n)$ using priority queue (See 7.2.2)
- **Can you reduce it to $o(n^2)$?**

K-Means

- An example of a point-assignment based clustering
  - Initially choose k points that are likely to be in different clusters
  - Make these points the centroids of their clusters
  - For each remaining point p DO
    - Find the centroid to which p is closest
    - Add p to the cluster of that centroid
    - Adjust the centroid of that cluster to account for p
  - Optional: reassign all points based on the new centroids. Repeat as long as there is any change in assignment.

How to select the K centers?