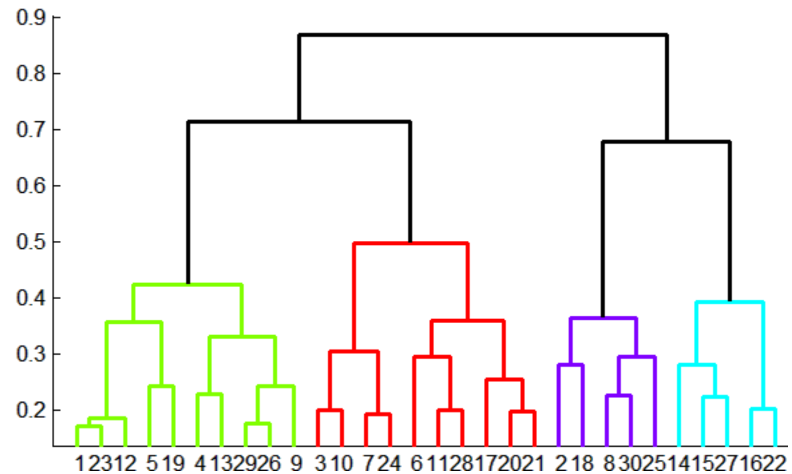
Clustering

Hierarchical Clustering

Mining of Massive Datasets
Leskovec, Rajaraman, and Ullman
Stanford University

Hierarchical Clustering

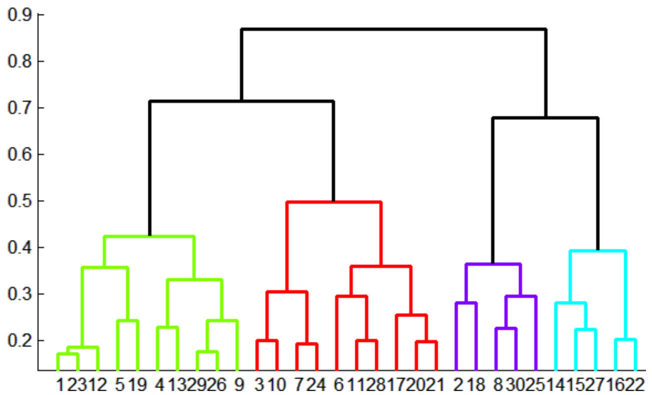
- **Hierarchical:**
  - **Agglomerative** (bottom up):
    - Initially, each point is a cluster
    - Repeatedly combine the two "nearest" clusters into one
  - **Divisive** (top down):
    - Start with one cluster and recursively split it



- This lecture: agglomerative approach
  - Same ideas can be used for divisive

Hierarchical Clustering

- **Key operation:**
  Repeatedly combine two nearest clusters

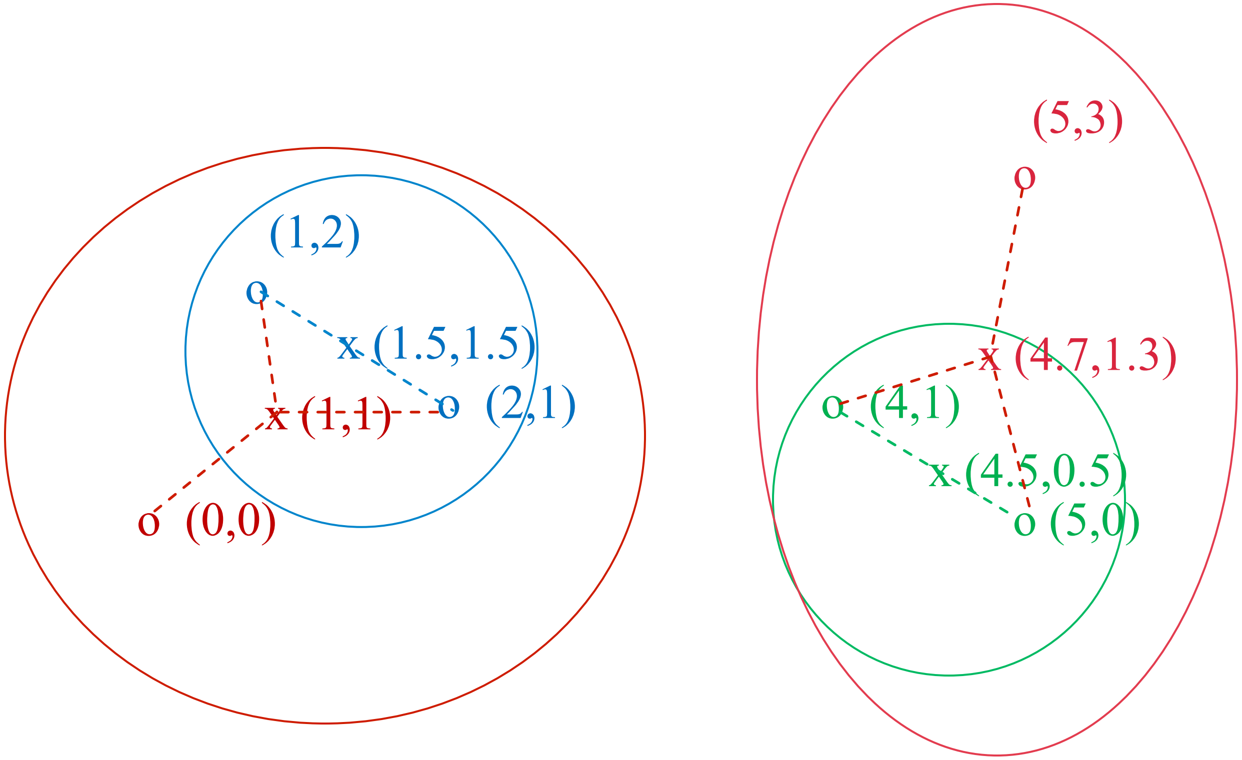


- **Three important questions:**
  - **1)** How do you represent a cluster of more than one point?
  - **2)** How do you determine the “nearness” of clusters?
  - **3)** When to stop combining clusters?

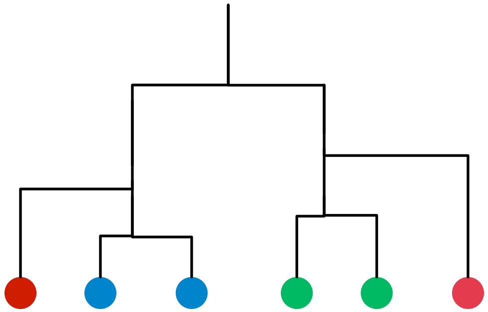
Euclidean Space

- **(1) How to represent a cluster of many points?**
  - How do you represent the location of each cluster, to tell which pair of clusters is closest?
  - Represent each cluster by its ***centroid*** = average of its points
- **(2) How to determine "nearness" of clusters?**
  - Measure cluster distances by distances of centroids

Example: Hierarchical clustering



Data:
o … data point
x … centroid



Dendrogram

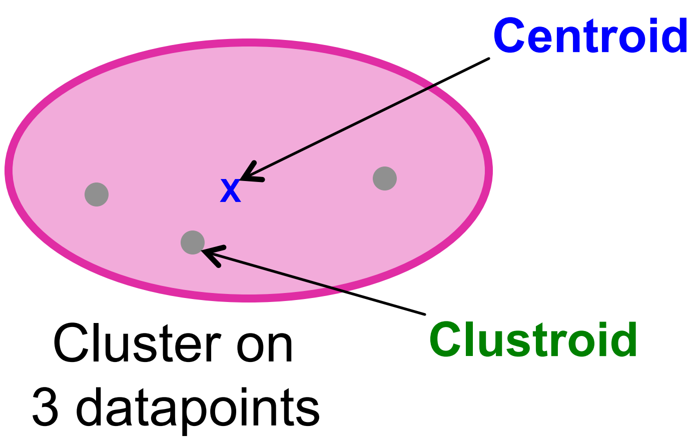
Non-Euclidean Case

- The only "locations" we can talk about are the points themselves
  - i.e., there is no "average" of two points

- **(1) How to represent a cluster of many points?** ***clustroid*** = (data)point "***closest***" to other points

- **(2) How do you determine the "nearness" of clusters?** Treat clustroid as if it were centroid, when computing intercluster distances

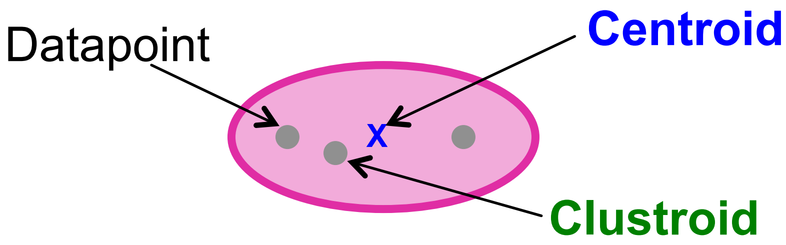
Clustroid



Centroid is the avg. of all (data)points in the cluster. This means centroid is an "artificial" point.

Clustroid is an **existing** (data)point that is "closest" to all other points in the cluster.

"Closest" Point?



- ***Clustroid*** = point "***closest***" to other points

- **Possible meanings of "closest":**
  - Smallest maximum distance to other points
  - Smallest average distance to other points
  - Smallest sum of squares of distances to other points

Termination condition

- (3) When do you stop combining clusters?
- Approach 1: Pick a number *k* upfront, and stop when we have *k* clusters
  - Makes sense when we know that the data naturally falls into *k* classes
- Approach 2: Stop when the next merge would create a cluster with low "cohesion"
  - i.e, a "bad" cluster

Cohesion

- **Approach 3.1: Diameter** of the merged cluster = maximum distance between points in the cluster

- **Approach 3.2: Radius** = maximum distance of a point from centroid (or clustroid)

- **Approach 3.3:** Use a **density-based approach**
  - Density = number of points per unit volume
  - E.g., divide number of points in cluster by diameter or radius of the cluster
  - Perhaps use a power of the radius (e.g., square or cube)

Implementation

- **Naïve implementation of hierarchical clustering:**
  - At each step, compute pairwise distances between all pairs of clusters, then merge
  - $O(N^3)$
- Careful implementation using priority queue can reduce time to $O(N^2 \log N)$
  - **Still too expensive for really big datasets that do not fit in memory**