Clustering

CompSci 590.03
Instructor: Ashwin Machanavajjhala

Duke UNIVERSITY

Clustering Problem

- 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 in some sense as close to each other as possible.

Example: Clustering News Articles

- Consider some vocabulary V = {v1, v2, …, vk}.
- Each news article is a vector (x1, x2, …, xk), where xi = 1 iff vi appears in the article
- Documents with similar sets of words correspond to similar topics

Example: Clustering movies (Collaborative Filtering)

- Represent each movie by the set of users who rated it.
- Each movie is a vector (x1, x2, …, xk), where xi is the rating provided by user i.
- Similar movies have similar ratings from the same sets of users.

Example: Protein Sequences

- Objects are sequences of {C, A, T, G}
- Distance between two sequences is the ***edit distance***, or the minimum number of inserts and deletes needed to change one sequence to another.
- Clusters correspond to proteins with similar sequences.

Outline

- Distance measures
- Clustering algorithms
  - K-Means Clustering
  - Hierarchical Clustering
- Scaling up Clustering Algorithms
  - Canopy Clustering

Distance Measures

- Each clustering problem is based on some notion of distance between objects or points
  - Also called similarity

- Euclidean Distance
  - Based on a set of m real valued dimensions
  - Euclidean distance is based on the locations of the points in the m-dimensional space
  - There is a notion of *average* of two points

- Non-Euclidean Distance
  - Not based on the location of points
  - Notion of average may not be defined

Distance Metric

- A distance function is a metric if it satisfies the following conditions

- $d(x,y) \geq 0$
- $d(x,y) = 0$ iff $x = y$
- $d(x,y) = d(y,x)$
- $d(x,y) \leq d(x,z) + d(z,y)$ *triangle inequality*

Examples of Distance Metrics

- Lp norm:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_i (x_i - y_i)^p \right)^{\frac{1}{p}}$$

- L2 norm = Distance in euclidean space
- L1 norm = Manhattan distance
- L∞ norm = maximum ($x_i - y_i$)

Examples of Distance Metrics

- Jaccard Distance:

Let A and B be two sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Examples of Distance Metrics

- Cosine Similarity:

$$cosine(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}$$

Examples of Distance Metrics

- Levenshtein distance a.k.a. Edit distance

  Minimum number of inserts and deletes of characters needed to turn one string into another.

Outline

- Distance measures
- Clustering algorithms
  - K-Means Clustering
  - Hierarchical Clustering
- Scaling up Clustering Algorithms
  - Canopy Clustering

K-Means

- A very popular *point assignment* based clustering algorithm
- Goal: Partition a set of points into k clusters, such that points within a cluster are closer to each other than point from different clusters.
- Distance measure is typically Euclidean
  - K-medians if distance measure does not permit an average

K-Means

- Input:
  - A set of points in *m* dimensions {x1, x2, …, xn}
  - The desired number of clusters K
- Output:
  - A mapping from points to clusters C: {1, …, m} → {1, …, K}

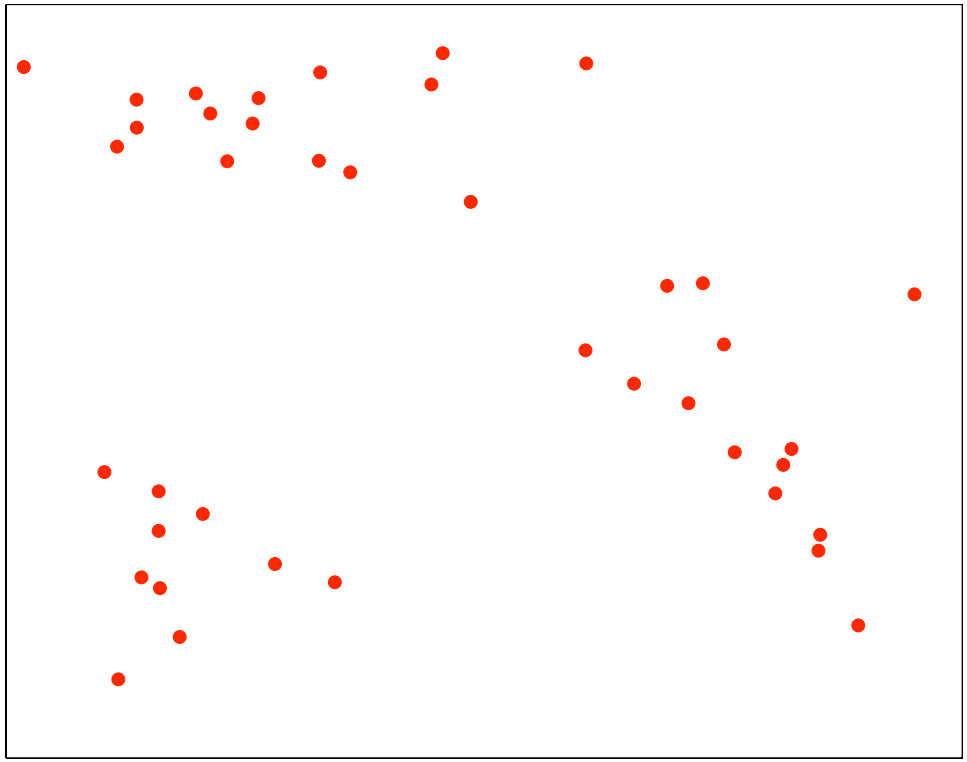
K-Means

- Input:
  - A set of points in *m* dimensions {x1, x2, …, xn}
  - The desired number of clusters K
- Output:
  - A mapping from points to clusters C: {1, …, m} → {1, …, K}

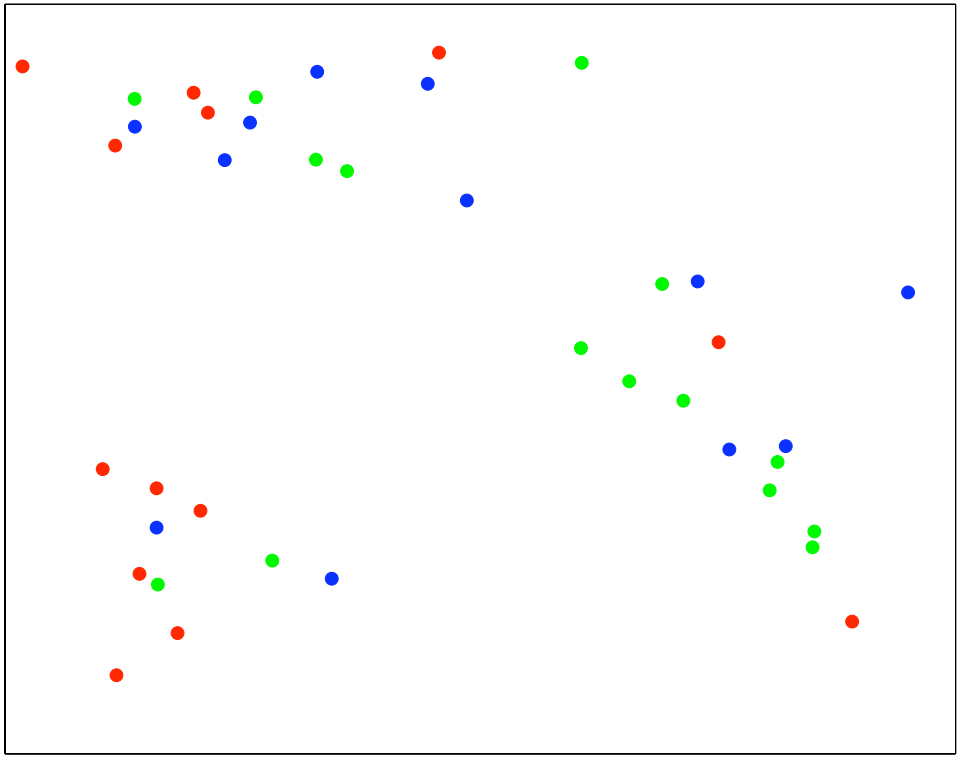
Algorithm:

- Start with an *arbitrary* C
- Repeat
  - Compute the centroid of each cluster
  - Reassign each point to the closest centroid
- Until C converges

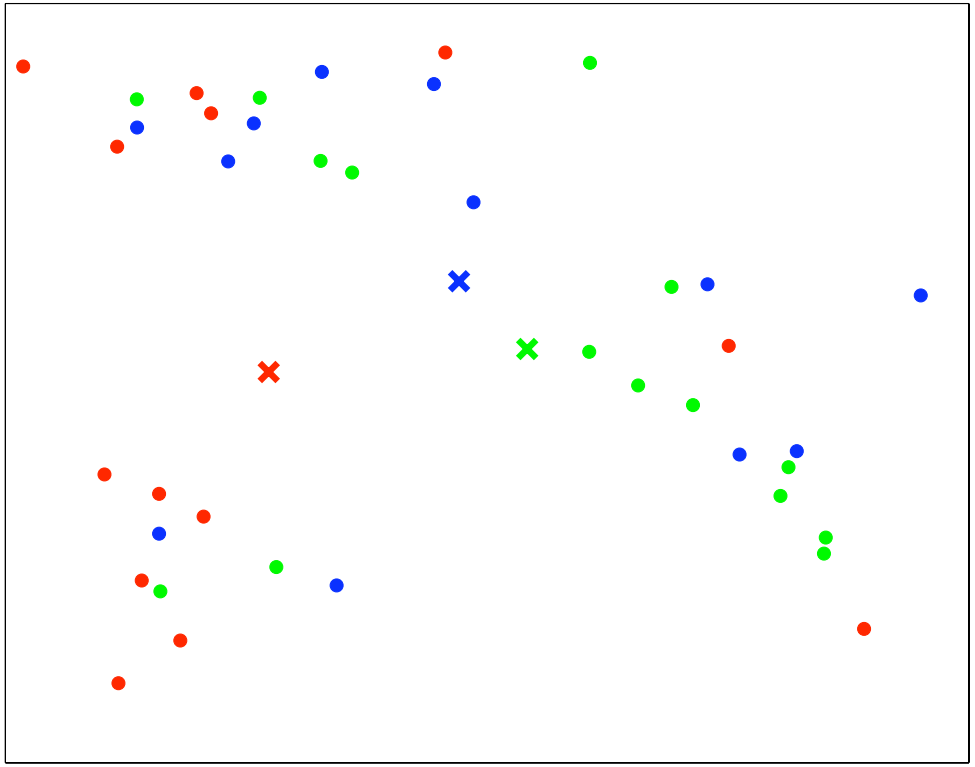
Example



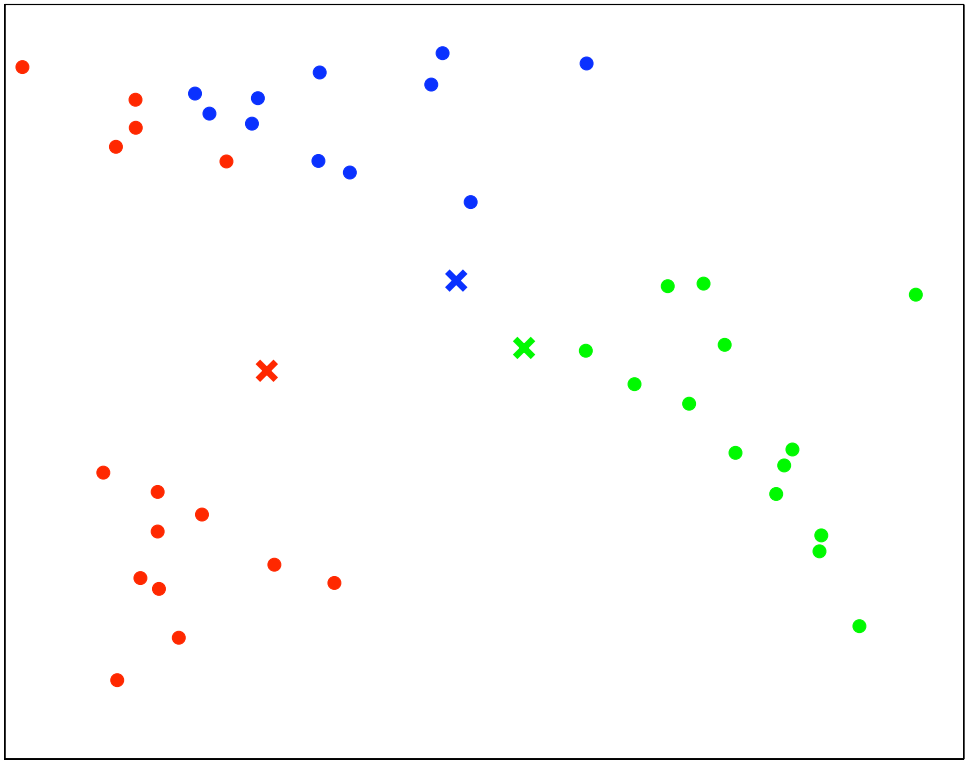
Initialize Clusters



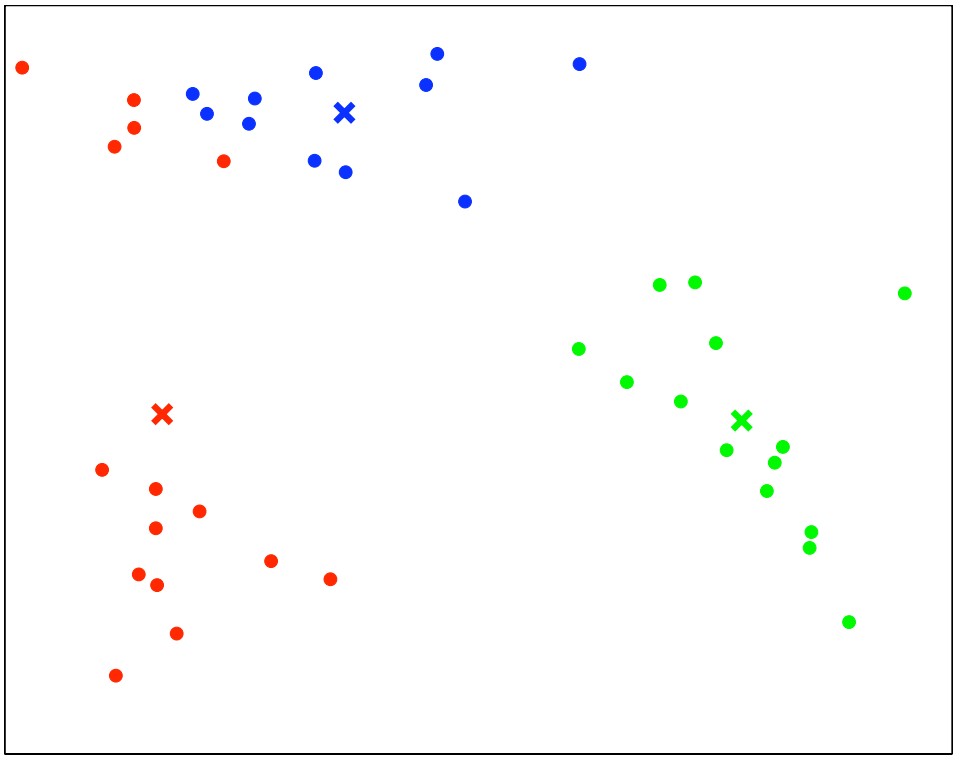
Compute Centroids



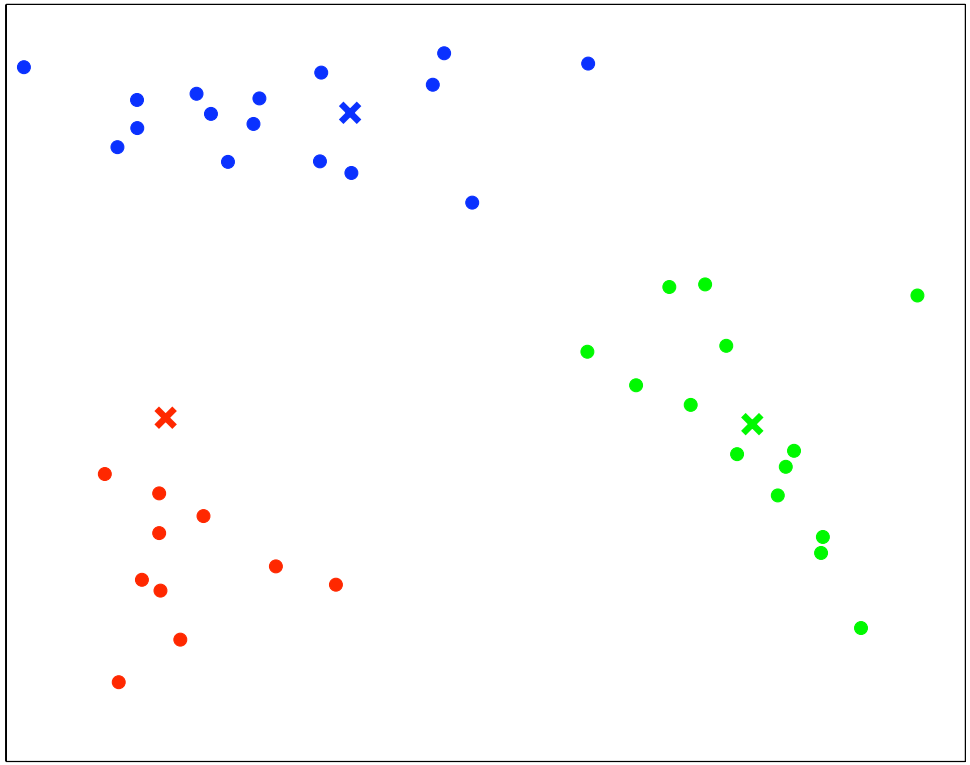
Reassign Clusters



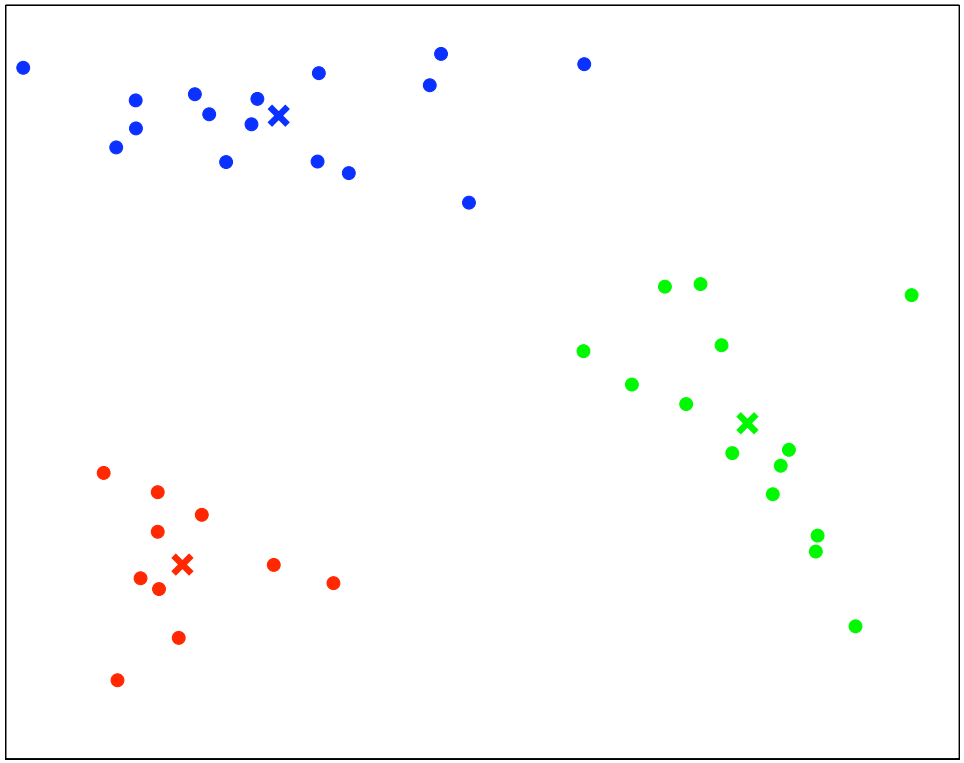
Recompute Centroids



Reassign Clusters



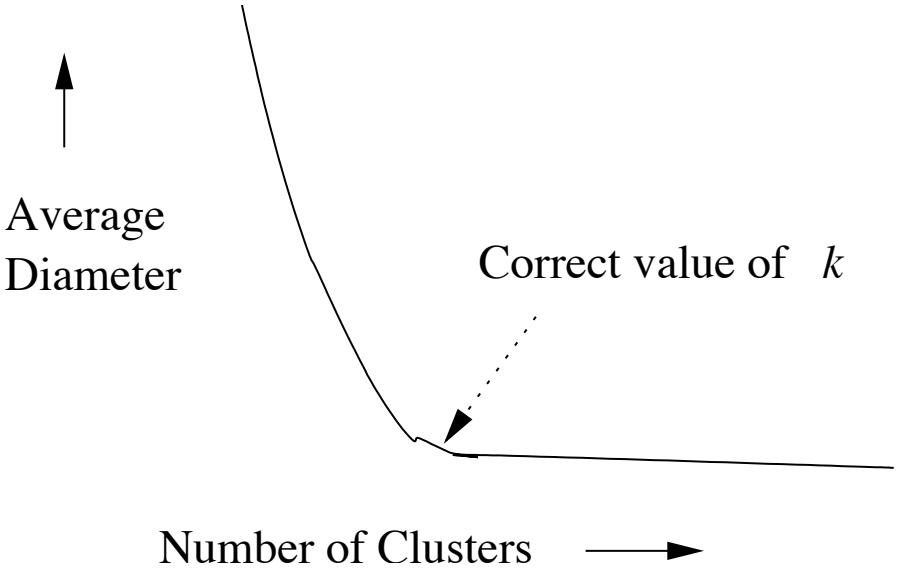
Recompute Centroids – Done!



Questions

- What is a good value for K?
- Does K-means always terminate?
- How should we choose initial cluster centers?

Determining K



- Small k: Many points have large distances to centroid
- Large k: No significant improvement in average diameter (max distance between any two points in a cluster)

K-means as an optimization problem

- Let ENCODE be a function mapping points in the dataset to {1…k}
- Let DECODE be a function mapping {1…k} to a point

$$min \sum_i (x_i - DECODE(ENCODE(x_i)))^2$$

- Alternately, if we write DECODE[j] = cj,
  we need to find an ENCODE function and k points c1, …, ck

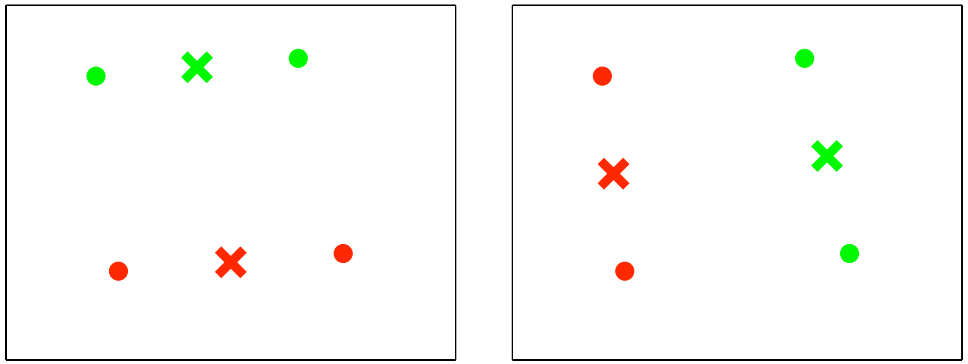
$$min \sum_i \left(x_i - c_{ENCODE(x_i)}\right)^2$$

K-means terminates

- Consider the objective function.
- There are finitely many possible clusterings (K^n)
- Each time we reassign a point to a nearer cluster, the objective decreases.
- Every time we recompute the centroids, the objective either stays the same or decreases.
- Therefore the algorithm has to terminate.

Local optima

- Depending on initialization K-means can converge to different local optima.



Initial Configuration

- Starting with a random assignment … cluster centroids will be close to the centroid of the entire dataset

- Farthest first heuristic
  - Choose first centroid to be a random point
  - Choose next centroid to be the point farthest away from the current set of centroids.

Outline

- Distance measures
- Clustering algorithms
  - K-Means Clustering
  - Hierarchical Clustering
- Scaling up Clustering Algorithms
  - Canopy Clustering

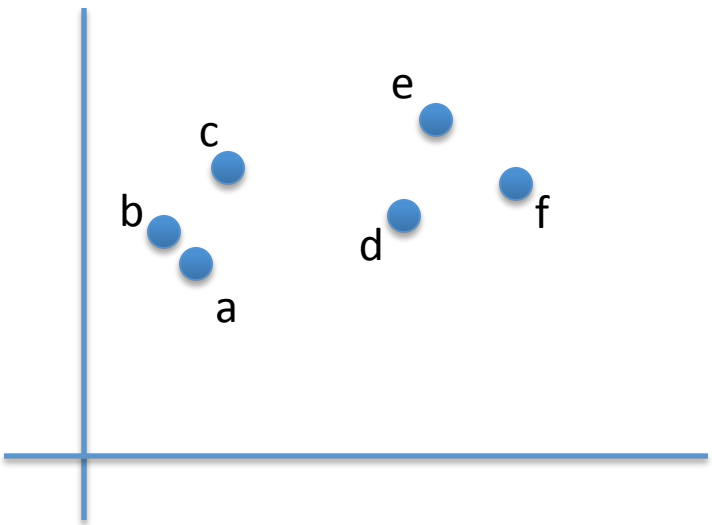
Hierarchical Clustering

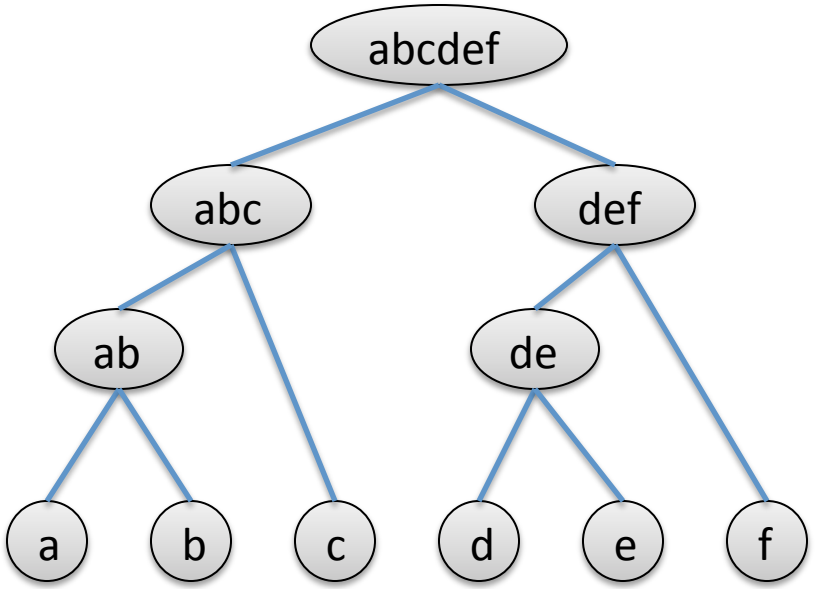
- Start with all points in their own clusters

- Repeat
  - Merge two clusters that are *closest to each other*
- Until (*stopping condition*)

Example

Distance metric: Euclidean distance





Distance between Clusters

- Different measures for distance between two clusters.

- Single Linkage

  $d(C1, C2) = \min_{x \text{ in } C1} \min_{y \text{ in } C2} d(x,y)$

- Average Linkage

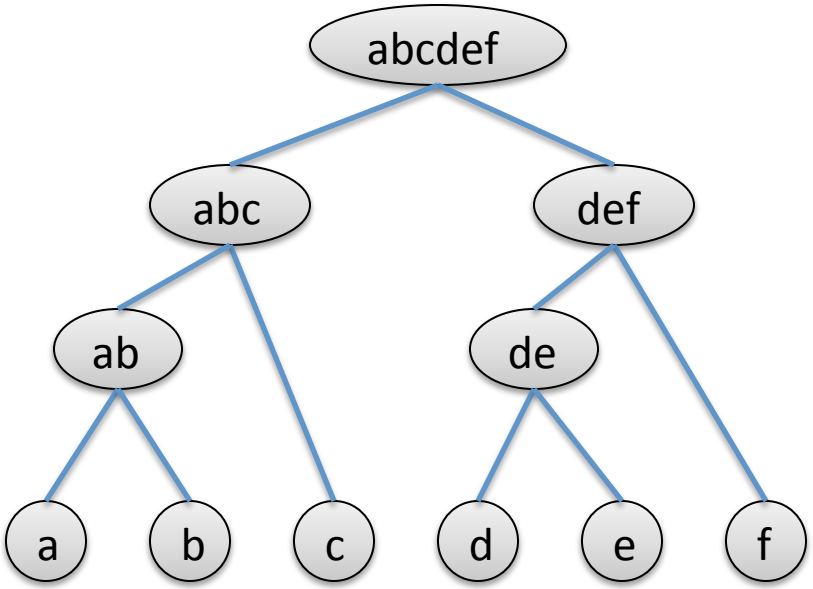
  $d(C1, C2) = \text{average}_{x \text{ in } C1,\, y \text{ in } C2} \{ d(x,y) \}$

- Complete Linkage

  $d(C1, C2) = \max_{x \text{ in } C1} \max_{y \text{ in } C2} d(x,y)$

Stopping Condition

- Dendogram



Stopping condition can depend on:

- Number of Clusters
- Distance between merging clusters
- Size of the largest cluster

Complexity

- Need to identify the closest clusters at each step
- Hence, need $\Omega(n^2)$ computation just to compute all the pairwise distances.
- We will see ways to speed up clustering next.

Outline

- Distance measures
- Clustering algorithms
  - K-Means Clustering
  - Hierarchical Clustering
- Scaling up Clustering Algorithms
  - Canopy Clustering

Scaling up Clustering

- Efficient clustering is possible when:
  - Small dimensionality
  - Small number of clusters
  - Moderate size data

- How to scale clustering when none of these hold?

Intuition behind Canopy Clustering

- Do not computing all $O(n^2)$ pairwise distances.

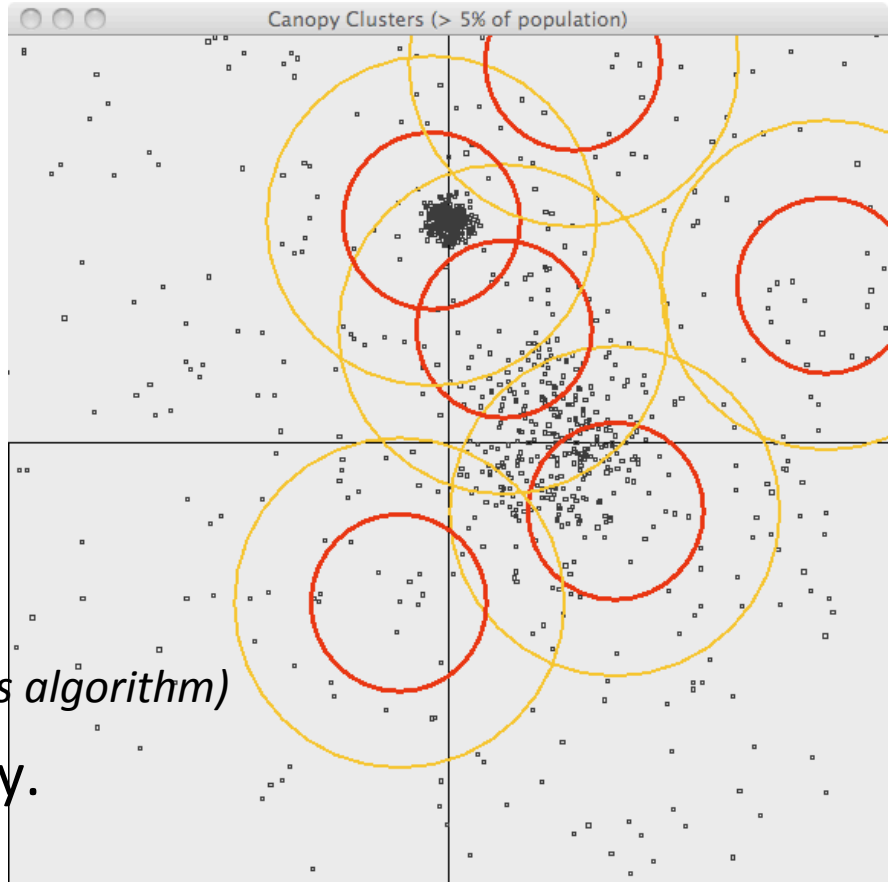
- For every point x, identify c(x) a small subset of points in the dataset which are most likely to be in the same cluster as x.

Canopy Clustering [McCallum et al KDD'00]

Input: Mentions M,
$d(x,y)$, a distance metric,
thresholds $T_1 > T_2$

Algorithm:

1. Pick a random element x from M
2. Create new canopy C_x using mentions y s.t. $d(x,y) < T_1$
3. Delete all mentions y from M s.t. $d(x,y) < T_2$ *(from consideration in this algorithm)*
4. Return to Step 1 if M is not empty.



Duke UNIVERSITY

Summary

- Clustering algorithms have a number of applications
- K-means and hierarchical clustering are popular techniques
- Canopy clustering helps scale clustering techniques