Clustering = Class discovery

COMP462/561: Computational Biology Methods

Fall 2016

M & W: 10:00 am – 11:30 am

*Based on Course Notes by Dr. Mathieu Blanchette

Motivation

Given: A collection of <u>unlabeled</u> samples $X_1 \ldots X_n$, where X_i represents the data for sample i

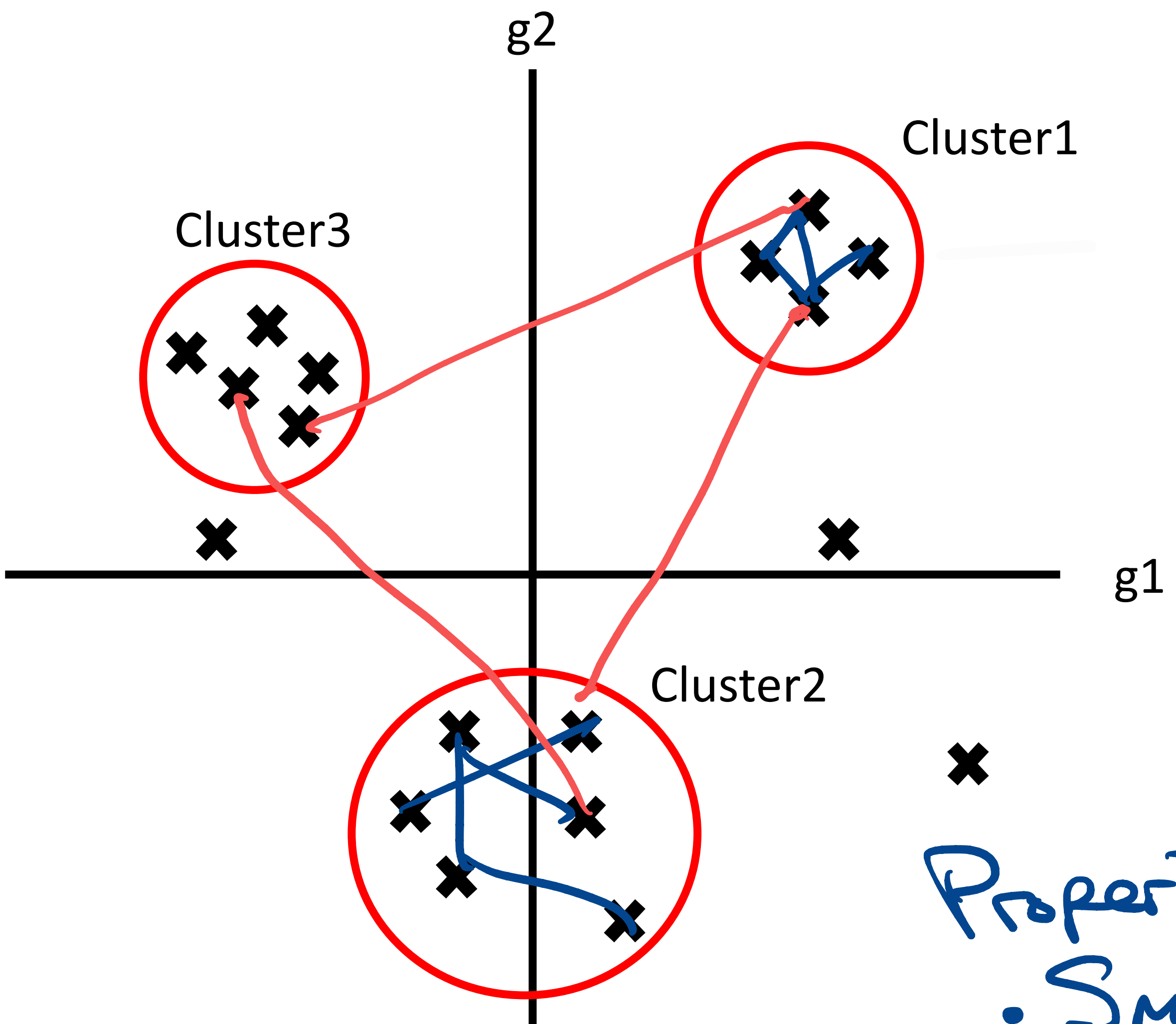
Goal: Partition samples into groups that are similar within themselves but dissimilar between

No information about class of each sample

	X_1	…	X_n
gene1	5		
gene2	2.1		
gene3	7.3		
…			
$gene_{k-1}$			
$gene_k$			

exp. profile of X_1

#genes = k = 2



Goal: Discover clusters

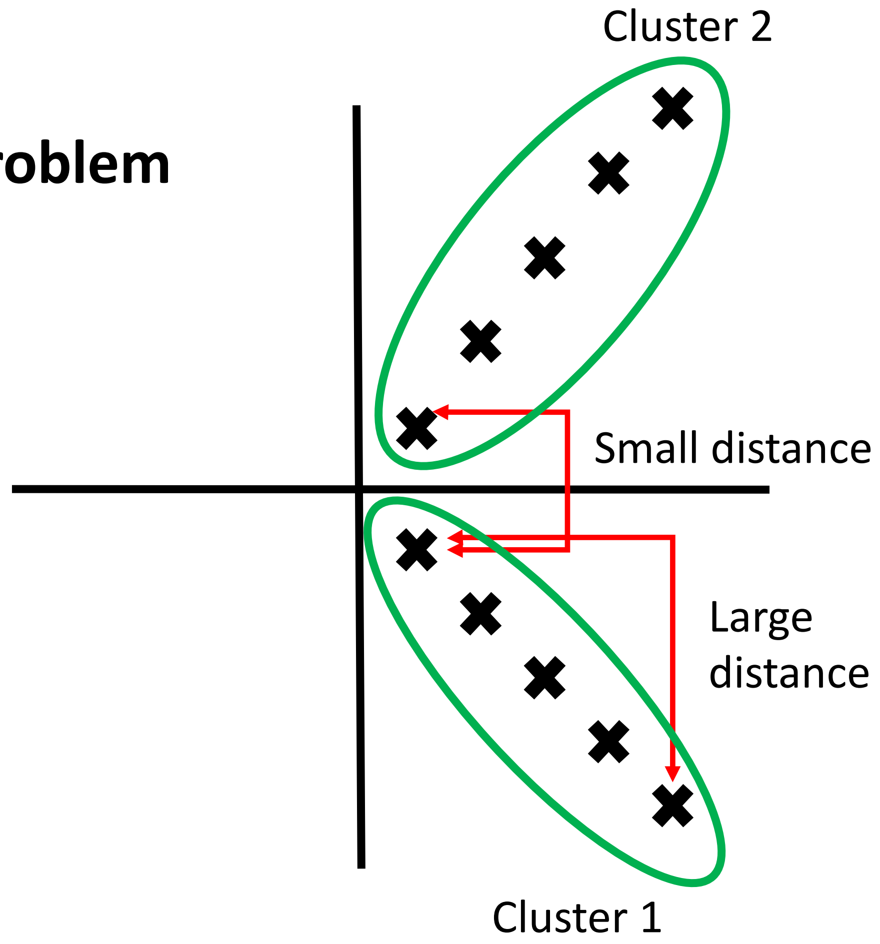
Properties:
- Small within-cluster distances
- Large inter-cluster distances

- All the clustering algorithms are **heuristic**
  - They don't guarantee the best clustering

Similarity (or Distance) Measures

Given: Two expression profiles, X_i and X_j

Problem



<u>**Euclidean Distance**</u>

$$d_E(X_i, X_j) = \sqrt{\sum_{g=1\ldots k} \left(X_{i,g} - X_{j,g}\right)^2}$$

Pearson Correlation Coefficient

Similarity Measure

$$Sim\left(X_i, X_j\right) = \frac{Cov(X_i, X_j)}{\sqrt{Var(X_i) \times Var(X_j)}}$$

$$= \frac{\sum(X_i(g) - \bar{X}_i)(X_j(g) - \bar{X}_j)}{\sqrt{(\sum(X_i(g) - \bar{X}_i)^2) \times (\sum(X_j(g) - \bar{X}_j)^2)}}$$

Max: +1

Uncorrelated: 0

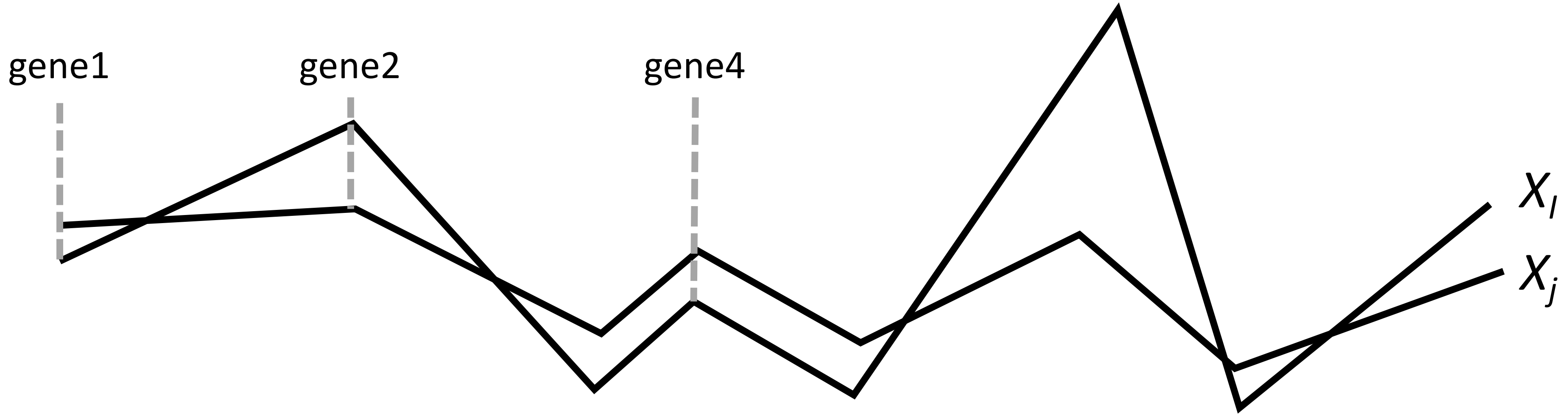
Min: -1

distance = 1 - correlation

Max: 2

Min: 0

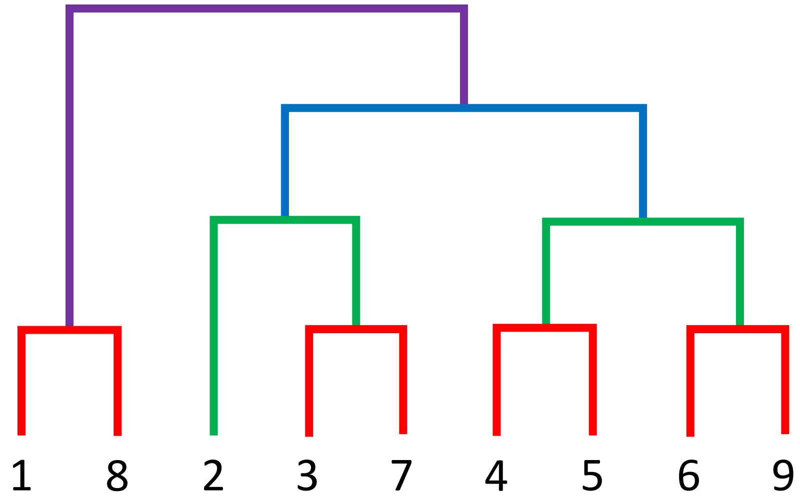
Pearson Correlation Coefficient Cont’d

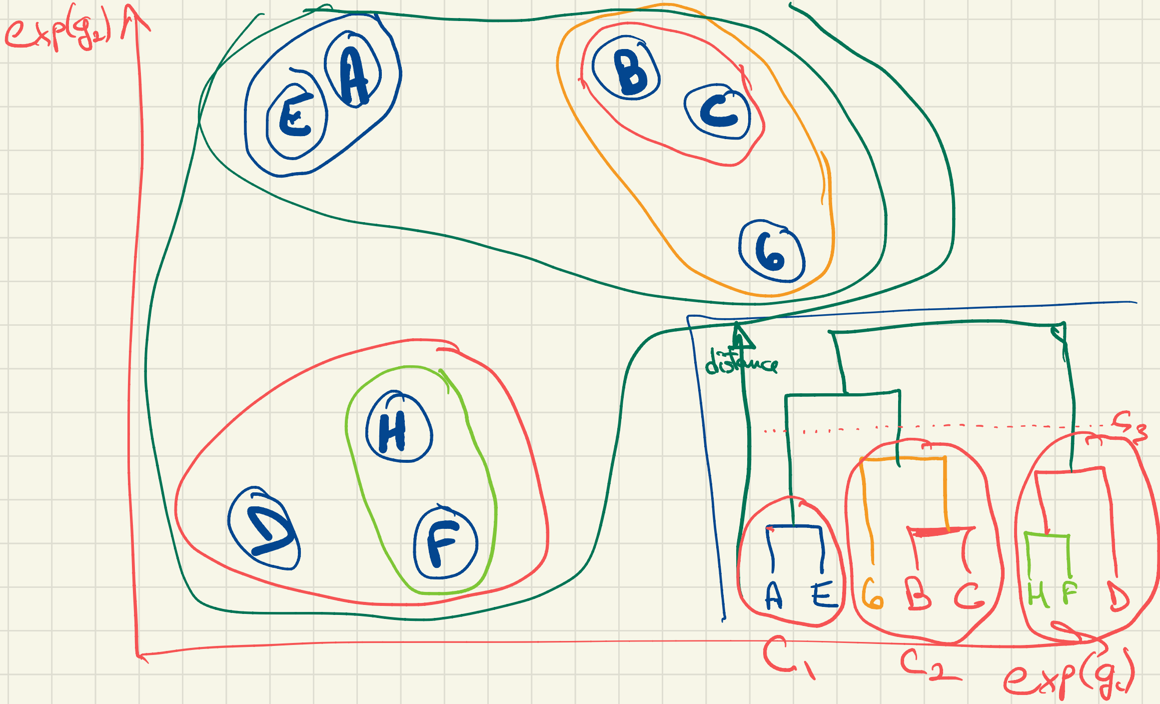


- Different expression level
  - But always goes in the same direction

Hierarchical Clustering

1. Start with each data point in its own cluster
2. Find the two clusters that are the closest and merge them
3. Repeat step two until all data points belong to a single cluster



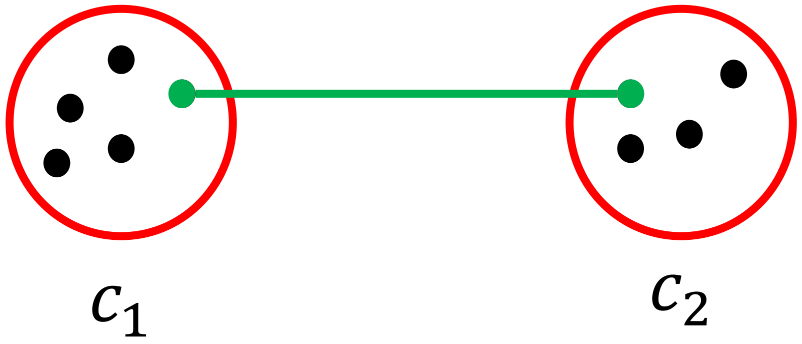


exp(g2)
E
A
B
C
G
H
D
F
distance
C3
A
E
G
B
C
H
F
D
C1
C2
exp(g1)

Measuring Similarity Between Clusters

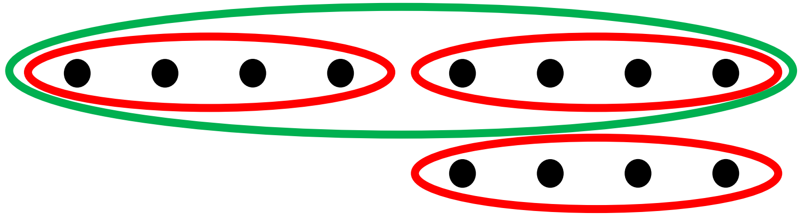
1) Single Linkage ~~approach~~ Clustering

$$Sim(c_1, c_2) = max_{x \in c_1, y \in c_2} \{sim(x, y)\}$$



Problem

- Given the following data points:

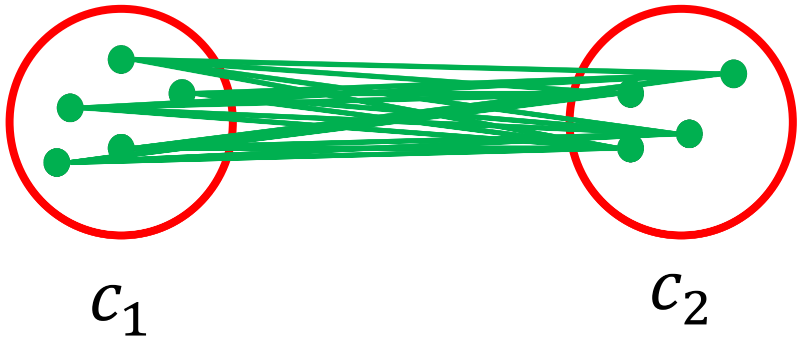


- Apply single linkage approach to clustering
- Get long and skinny clusters by having one point near the others
    - <u>Shouldn’t the two clusters on the right pair better together?</u>

Measuring Similarity Between Clusters

2) **Average linkage** Clustering

$$Sim(c_1, c_2) = \frac{1}{|c_1| \cdot |c_2|} \sum_{x \in c_1, y \in c_2} Sim(x, y)$$



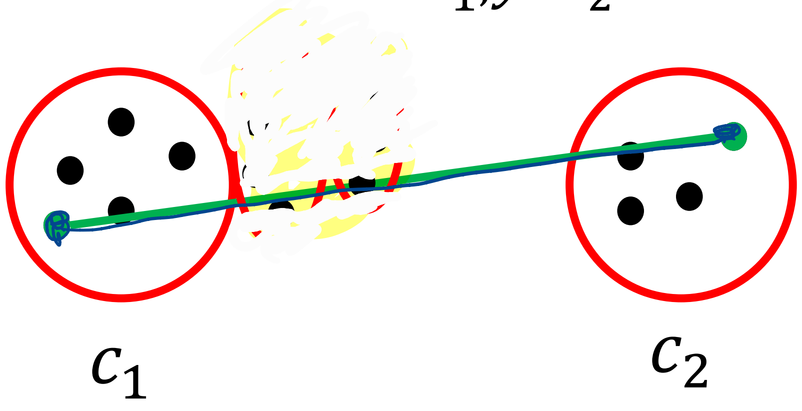
Take all pairs!

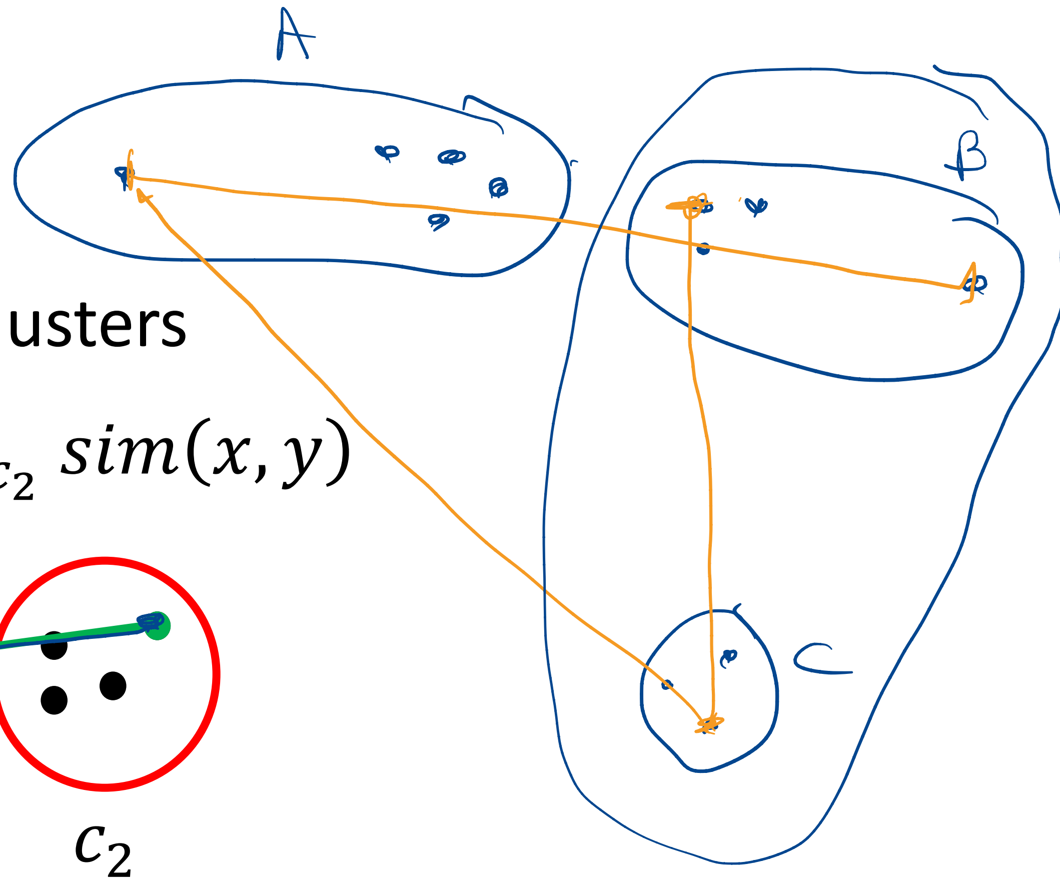
Measuring Similarity Between Clusters

3) Complete linkage Clustering

Makes very compact clusters

$$Sim(c_1, c_2) = min_{x \in c_1, y \in c_2}\ sim(x, y)$$

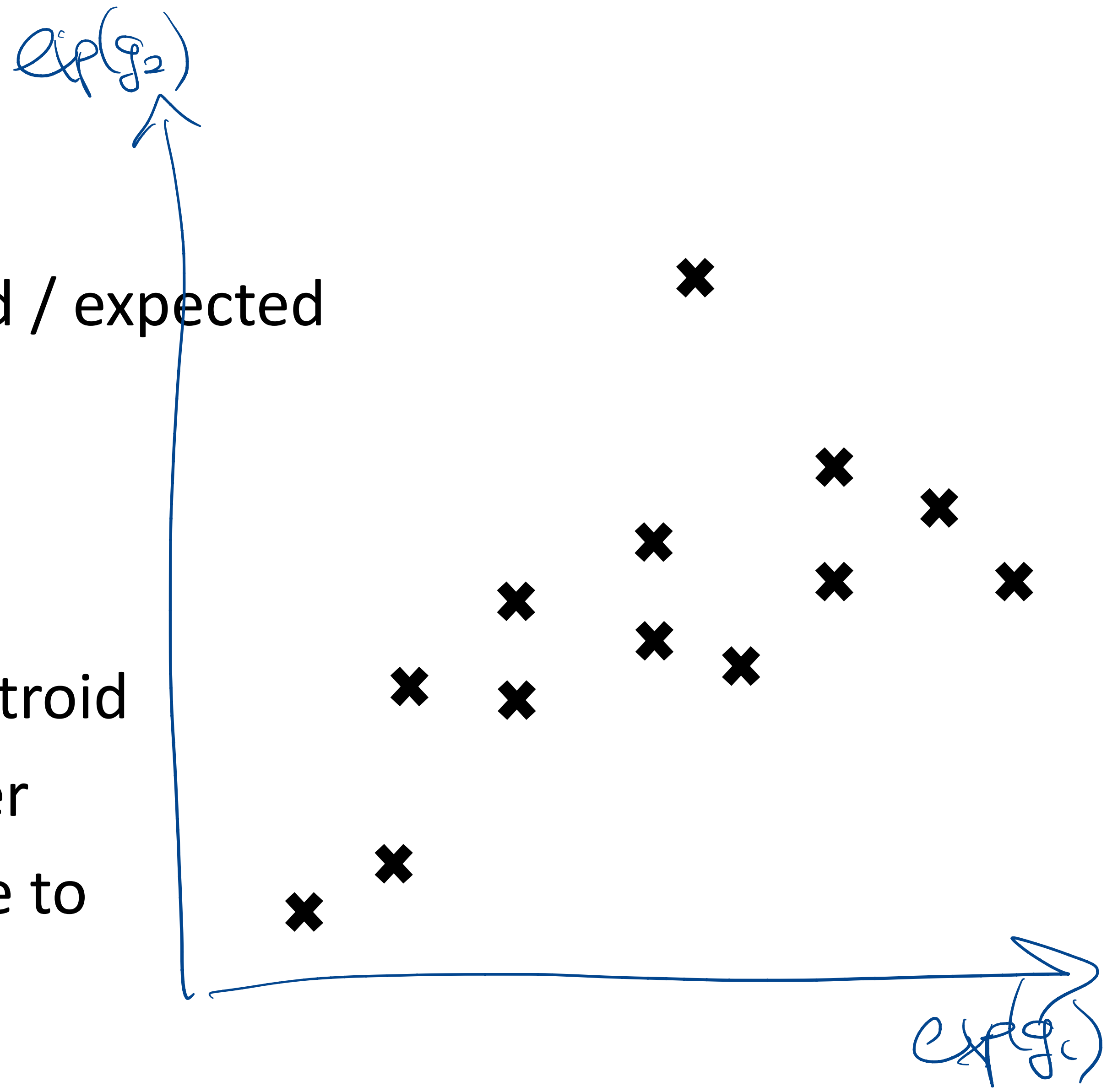




K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

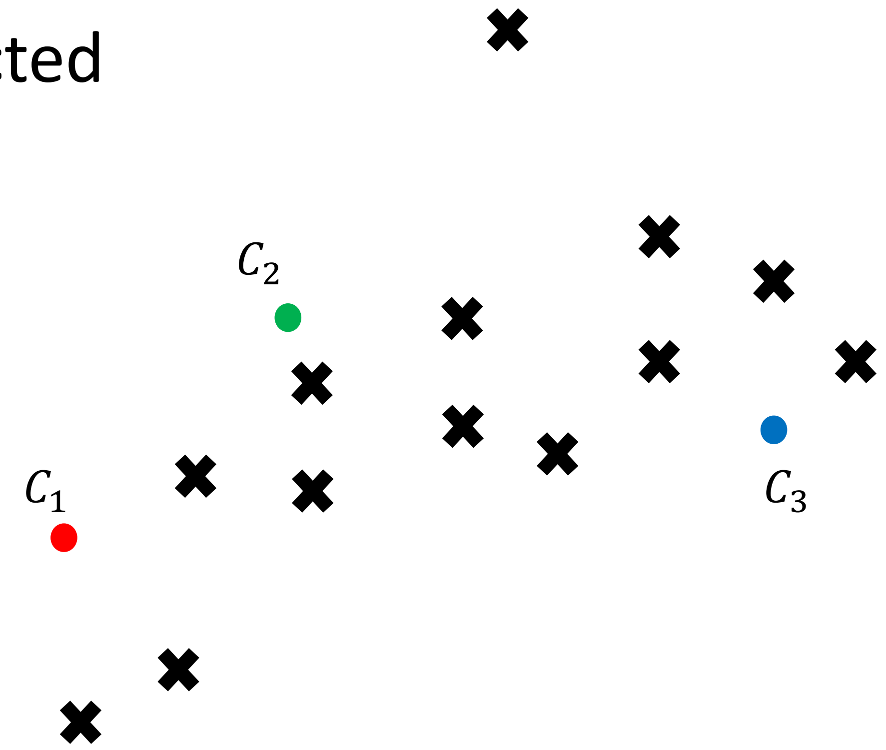
1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

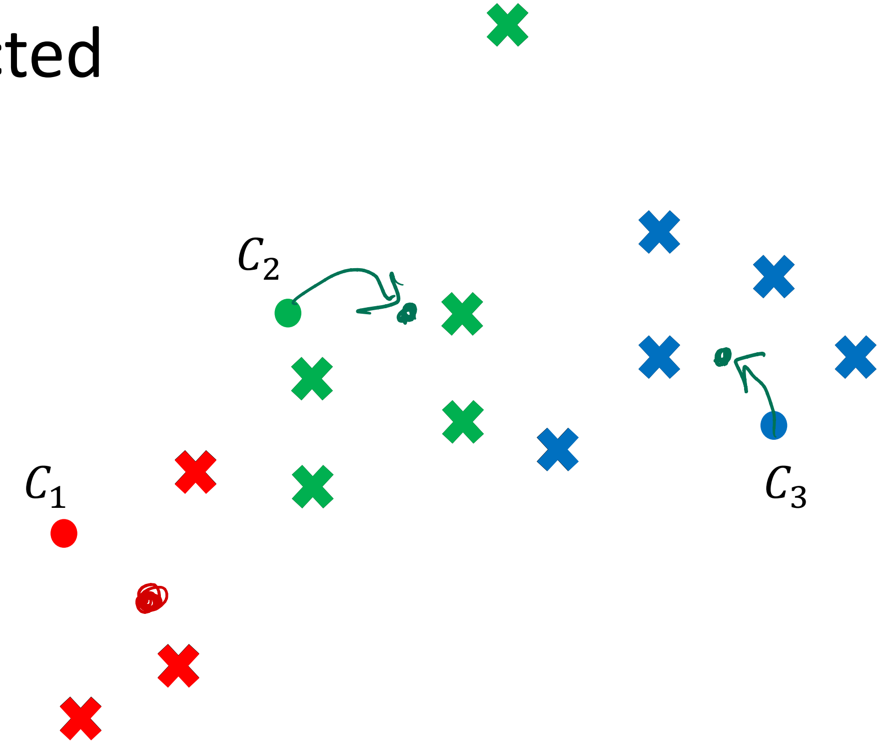
1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

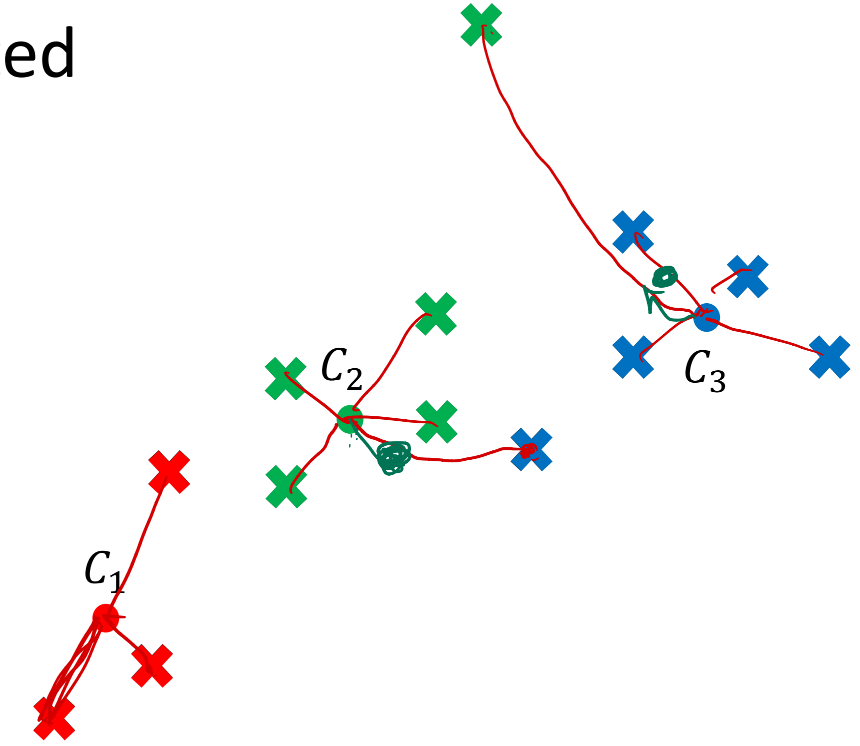
1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

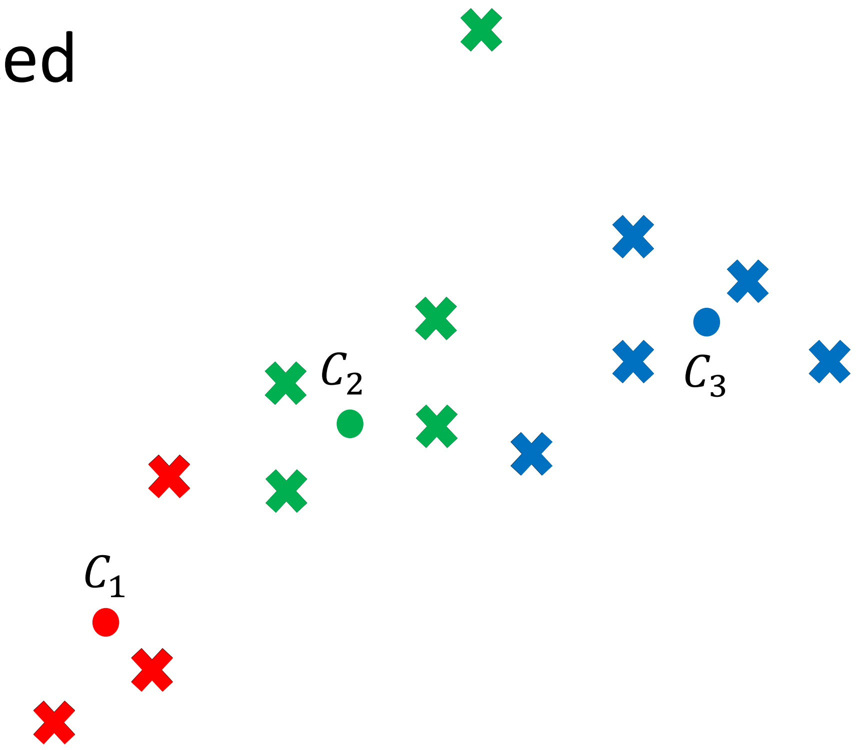
1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

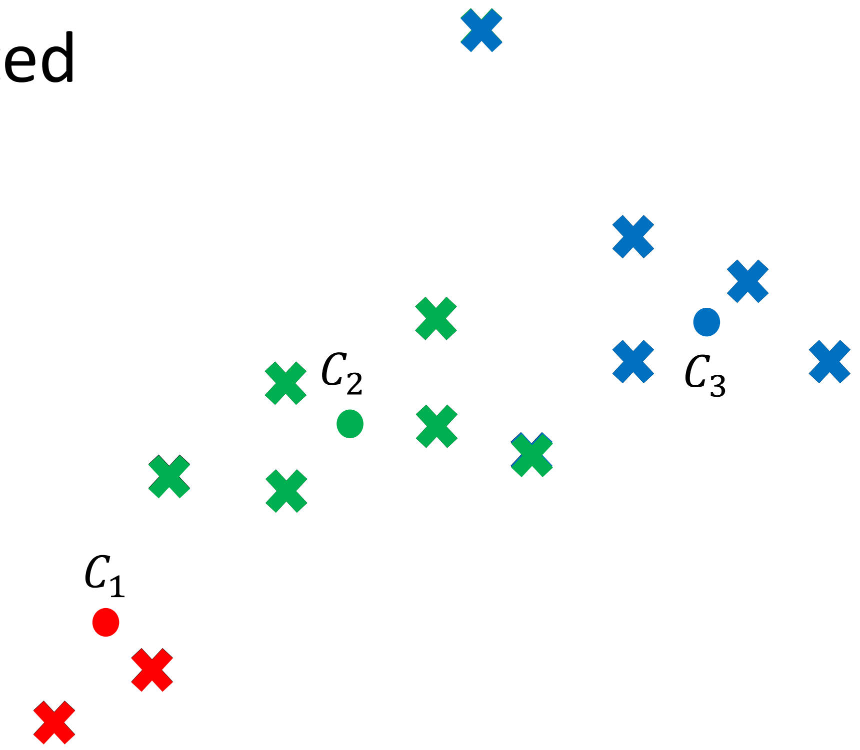
1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- *'k'* is the number of clusters desired / expected
- Each cluster has a centroid

1. Randomly choose k centroids

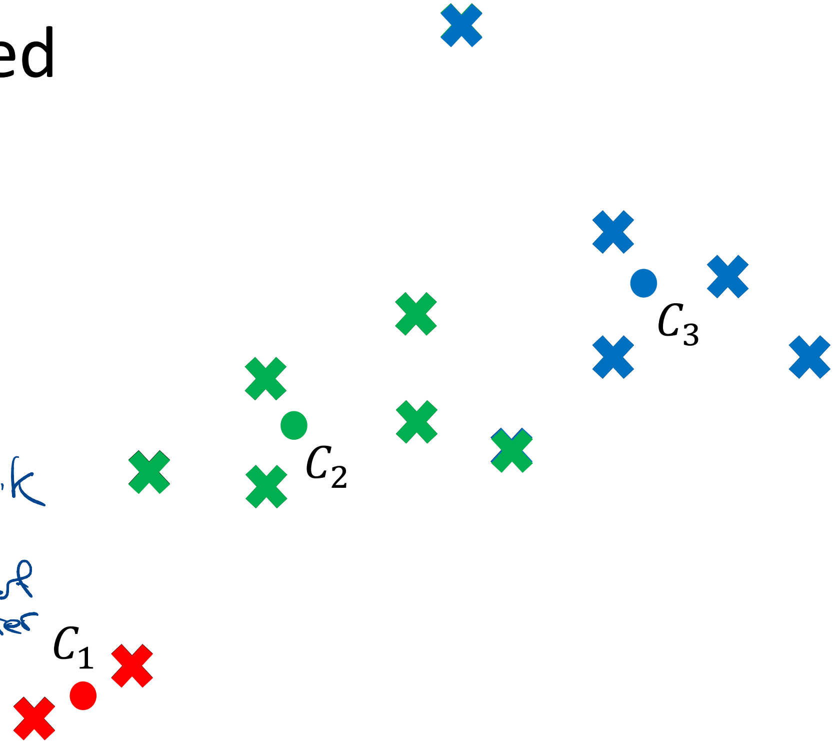
   n k

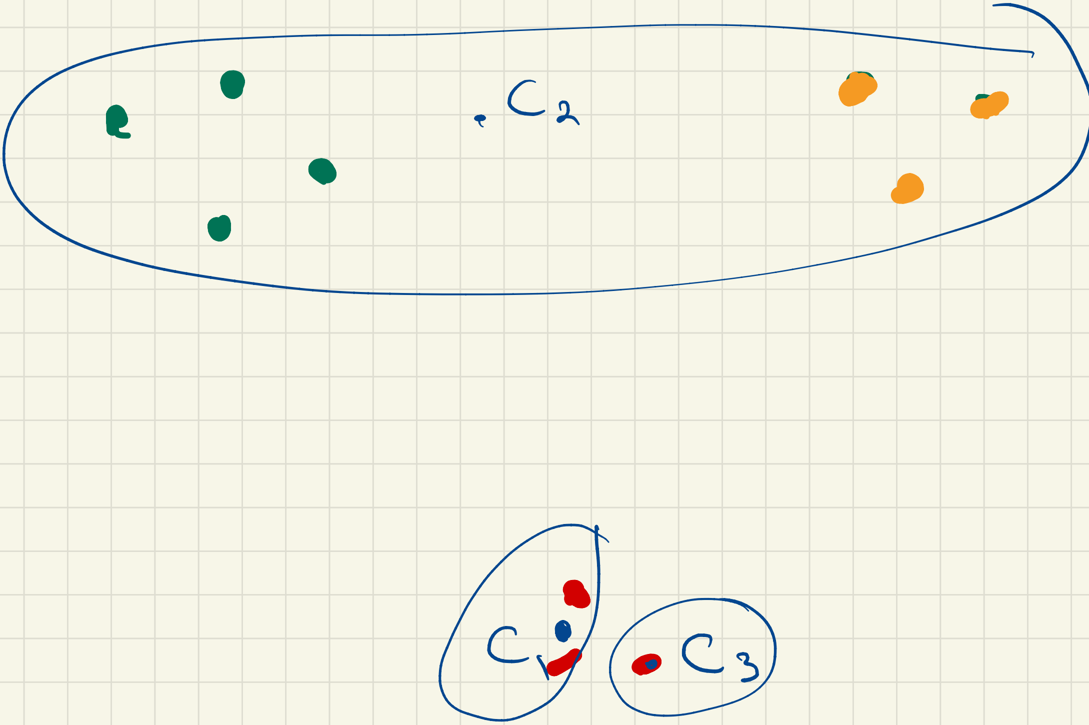
2. Assign data points to nearest centroid → $n \cdot k$
3. Move centroid to center of cluster

   $k \cdot$ size of cluster

   $\leq k \cdot n$

4. Repeat 2-4. Stop when no change to data point assignment





C_2
C_1
C_3

Running Time

Hierarchical cluster (avg linkage)

① Calculate all cluster-to-cluster distances

For: m clusters of k points

$$m^2 \cdot k^2$$

Repeat step 1 $n-1$ times

$\Rightarrow O(n^5)$ (probably $O(n^3)$)

K-means

One iteration $= O(n \cdot k + k \cdot n) = O(n \cdot k)$

Total $= O(n \cdot k \cdot I)$

$I =$ # of iteration $\approx O(\sqrt{n})$

Cluster Validation

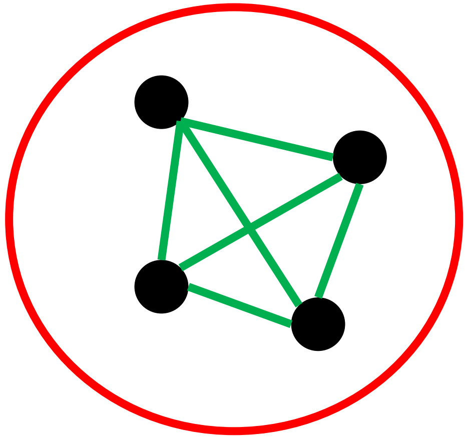
- **Cohesion:** measures how closely related data points in a cluster are (i.e., within cluster Sum of Squares [WSS])

$$WSS = \sum_{i} \sum_{x \in c_i} \|x - m_i\|^2$$

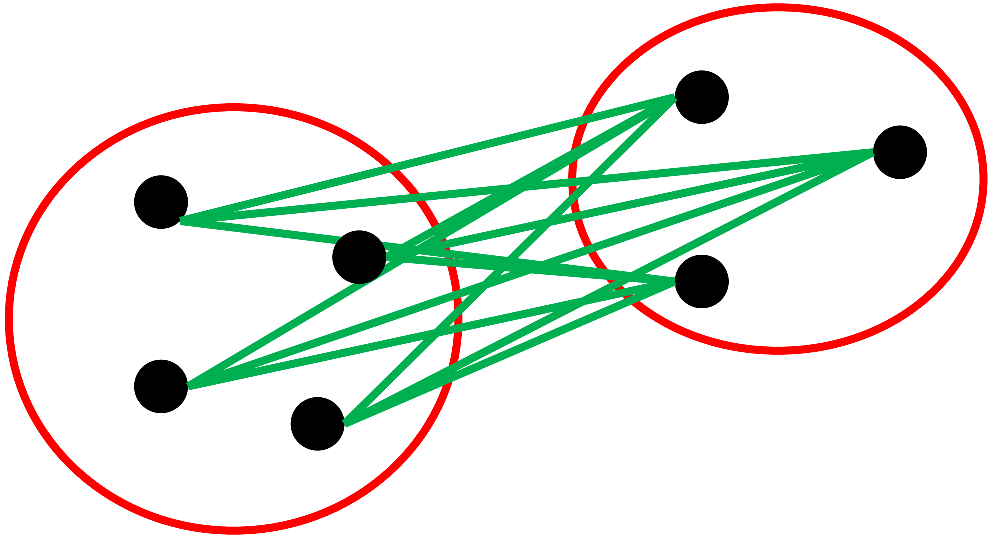
- **Separation:** measures how distinct or well-separated a cluster is from others (i.e., between cluster Sum of Squares [BSS])

$$BSS = \sum_{i} \sum_{j} |c_i| \cdot |c_j| \cdot \|m_i - m_j\|^2$$

Cohesion and Separation



Cohesion



Separation