GMMs, Hierarchical Clustering

Lecture 9:
Gaussian Mixture Models, Hierarchical Clustering

ECE/CS 498 DS
Professor Ravi K. Iyer
Department of Electrical and Computer Engineering
University of Illinois

Announcements

- MP 1 final checkpoint **due tomorrow Feb 20th @ 11:59 PM on Compass2G**
  - One submission per group, consisting of
    - Single ipynb for all tasks
    - Single PDF with results for all tasks (template has been provided)
  - **Presentation signup link is live**: https://docs.google.com/spreadsheets/d/14braJUAaud3y4kcg6I1N1fTRBxZBis6sutxqxTpWsKU/edit#gid=0
- Discuss section this week (2/21) is cancelled due to MP 1 presentations
- HW 2 will be released this upcoming Mon Feb 24
  - Covers Bayesian networks and inferencing
- MP 2 will be released this upcoming Mon Feb 24
  - Uses health data collected from the gut microbiome
- Midterm exam will take place on **Wed March 11th**

Gaussian Mixture Models

Expectation-Maximization: Motivation

- Clustering data points using MAP or maximum likelihood rules is very difficult when there are **latent (hidden / unobservable) variables**
  - Latent variables interact with the dataset but are not directly observed/known
  - In clustering, these latent variables are usually parameters of the clusters we are trying to determine (e.g. centroid locations in k-means, mean and standard deviation in Gaussian clustering)
- **Expectation Maximization** is an iterative solution to this problem
  - General procedure:
    - (1) Initialization Step: "guess" latent variables (e.g. cluster parameters)
    - (2) Expectation Step: optimize model to fit the data using the currently known latent variables
    - (3) Maximization Step: optimize the parameters using the current model
    - (4) Repeat steps (2)-(3) until convergence

https://machinelearningmastery.com/expectation-maximization-em-algorithm/

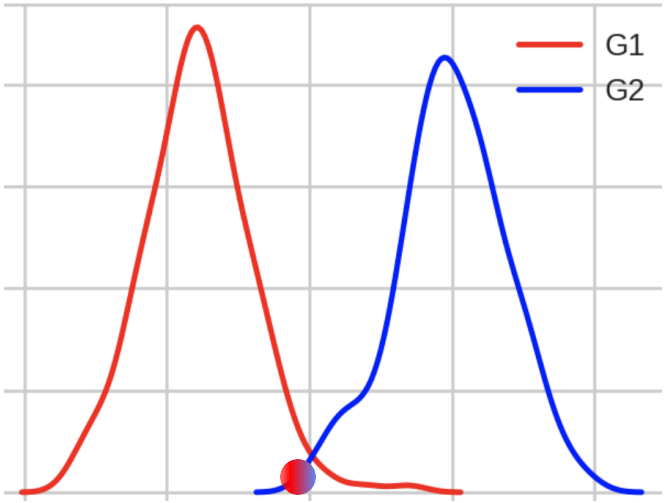
Soft Clustering: Mixture Model: Clusters may Overlap

Given:

- Data points/observations: $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots$

Model:

- There is a set of *K* probability distributions
  - Each distribution represents a cluster
  - Each distribution is described by certain parameters
  - Clusters may overlap
  - Find strengths of association between clusters and data instances
  - Discover the parameters of the distribution e.g. mean and variance
- Each data point is sampled from one of several distributions
  - $p(\boldsymbol{x_i}|b)$: Likelihood probability (density) that an instance $\boldsymbol{x_i}$ takes certain feature values given that it is from cluster b
  - $P(b|\boldsymbol{x_i})$: Posterior probability that an instance belongs to cluster b given that its features are $\boldsymbol{x_i}$



Problem:

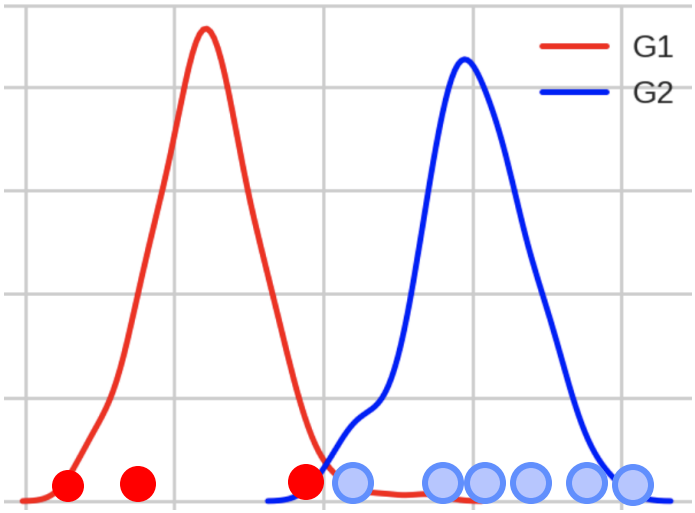
- Find parameters of the *K* distributions
- Find the posterior probabilities for each point

Expectation Maximization

- Automatically discover all the parameters for the K sources

GMM Example: Find parameters

- Observations: $x_1, x_2, \ldots, x_N$
  - Each observation has 1 feature (1-dimension)
- Data is sampled from one of two Gaussian distributions (K=2)
  - Cluster r: (μ_r, σ_r^2)
  - Cluster b: (μ_b, σ_b^2)



- **Estimation**: If source (cluster) of each observation is known, it is trivial to estimate (μ_r, σ_r^2) and (μ_b, σ_b^2)

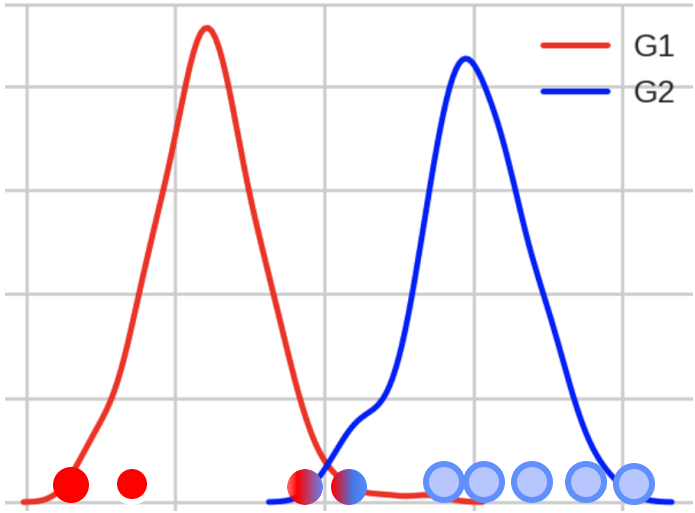
$$\mu_r = \frac{\sum_{i=1}^{N} x_i \mathbb{I}\{x_i \sim r\}}{\sum_{i=1}^{N} \mathbb{I}\{x_i \sim r\}} \qquad \sigma_r^2 = \frac{\sum_{i=1}^{N} (x_i - \mu_a)^2 \mathbb{I}\{x_i \sim r\}}{\sum_{i=1}^{N} \mathbb{I}\{x_i \sim r\}}$$

$$\mu_b = \frac{\sum_{i=1}^{N} x_i \mathbb{I}\{x_i \sim b\}}{\sum_{i=1}^{N} \mathbb{I}\{x_i \sim b\}} \qquad \sigma_b^2 = \frac{\sum_{i=1}^{N} (x_i - \mu_b)^2 \mathbb{I}\{x_i \sim b\}}{\sum_{i=1}^{N} \mathbb{I}\{x_i \sim b\}}$$

where $\mathbb{I}\{x_i \sim r\} = 1$ if x_i was sampled from cluster r and 0 otherwise.

GMM Example: Find posterior

- Observations: $x_1, x_2, \ldots, x_N$
  - Each observation has 1 feature (1-dimension)
- Data is sampled from one of two Gaussian distributions (K=2)
  - Cluster a: (μ_a, σ_a^2)
  - Cluster b: (μ_b, σ_b^2)



- If the distribution and its parameters are known, estimate where the point is likely to come from using Bayes rule

$$P(b|x_i) = \frac{p(x_i|b)P(b)}{p(x_i|b)P(b) + p(x_i|r)P(r)}$$

Posterior probability of distribution b given sample x_i

$$p(x_i|b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i-\mu_b)^2}{2\sigma_b^2}\right)$$

Probability density of observing x_i when sampled from distribution b

Expectation Maximization

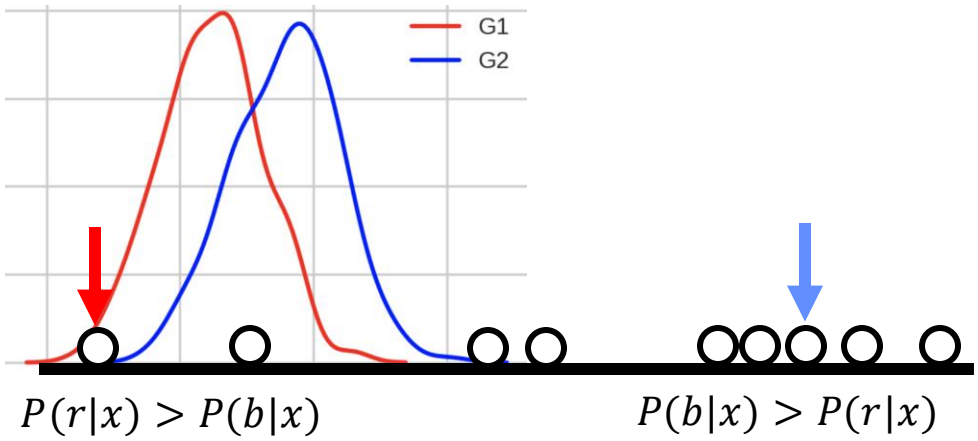
- **What if neither the source nor the distribution parameters are known?**

- **Chicken and Egg problem**
  - Need (μ_b, σ_b^2) and (μ_r, σ_r^2) to guess source of points
  - Need to know source to estimate (μ_b, σ_b^2) and (μ_r, σ_r^2)
  - Use **Expectation Maximization (EM)** algorithm

- **EM Algorithm**
  - Start with **two randomly placed Gaussians** (μ_b, σ_b^2) and (μ_r, σ_r^2)
  - For each x_i, calculate $P(b|x_i)$ and $P(r|x_i) = 1 - P(b|x_i)$
  - **Remember it does not assign the point but says here is the probability that it came from the red cluster or from the blue cluster (Soft assignment)**
  - Adjust (μ_b, σ_b^2) and (μ_r, σ_r^2) to fit points most likely belonging to them

GMM Example: EM in action

- Start with **two randomly placed Gaussians** (μ_b, σ_b^2) and (μ_r, σ_r^2)
- **Expectation step (E)**: Assign posterior probabilities to each sample x_i



- Let b_i be the posterior probability of sample x_i belonging to cluster b

$$b_i = P(b|x_i) = \frac{p(x_i|b)P(b)}{p(x_i|b)P(b) + p(x_i|r)P(r)}$$

$$p(x_i|b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i-\mu_b)^2}{2\sigma_b^2}\right)$$

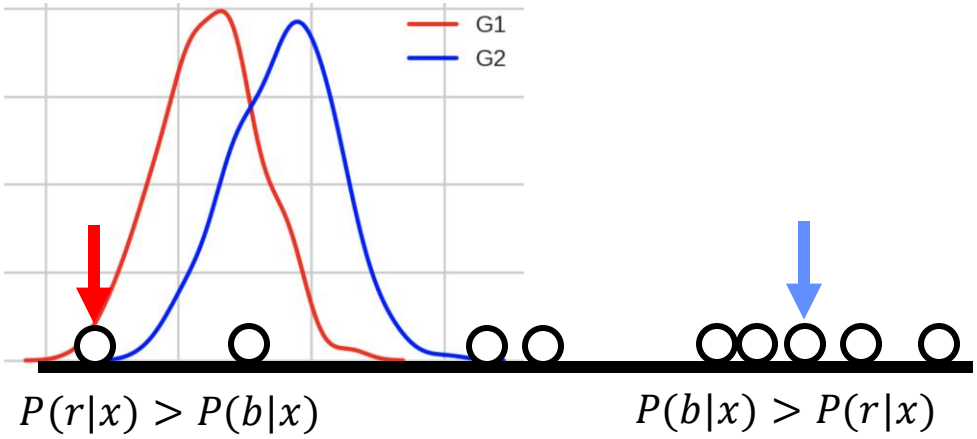
Probability density of observing x_i when sampled from distribution b

- Similarly, let r_i be the posterior probability of sample x_i belonging to cluster r

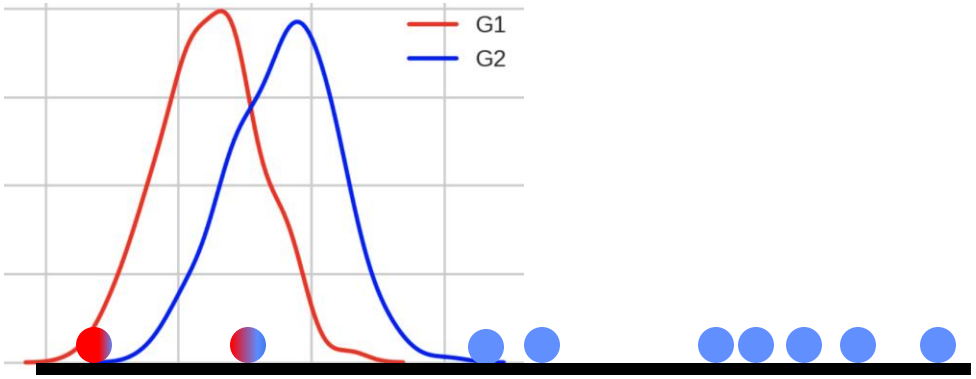
$$r_i = 1 - b_i$$

GMM Example: EM in action

Before assigning posterior probabilities b_i and r_i



After assigning posterior probabilities b_i and r_i



GMM Example: EM in action

- **Maximization step (M)**: Update the distribution parameters (re-estimation)
- Take **weighted average of the samples**
  - Weight is the posterior probability of that sample
- Similar to previous estimation, but with $\mathbb{I}\{x_i \sim b\}$ replaced by $P(b|x_i)$
  - $P(b|x_i)$ gives how likely it is that the cluster is b given the sample x_i
  - Therefore, x_i's contribution in re-estimating the parameters for b is $b_i = P(b|x_i)$

$$\mu_b = \frac{b_1 x_i + b_2 x_2 + \cdots + b_N x_N}{b_1 + b_2 + \cdots + b_N} = \frac{\sum_{i=1}^{N} b_i x_i}{\sum_{i=1}^{N} b_i}$$

Mean is simply weighted average of samples

$$\mu_r = \frac{\sum_{i=1}^{N} r_i x_i}{\sum_{i=1}^{N} r_i}$$

$$\sigma_b^2 = \frac{b_1 (x_1 - \mu_b)^2 + b_2 (x_2 - \mu_b)^2 + \cdots + b_N (x_N - \mu_b)^2}{b_1 + b_2 + \cdots + b_N}$$

$$= \frac{\sum_{i=1}^{N} b_i (x_i - \mu_b)^2}{\sum_{i=1}^{N} b_i}$$

Variance is weighted sum of square distances of samples from the distribution mean

$$\sigma_r^2 = \frac{\sum_{i=1}^{N} r (x_i - \mu_r)^2}{\sum_{i=1}^{N} r_i}$$

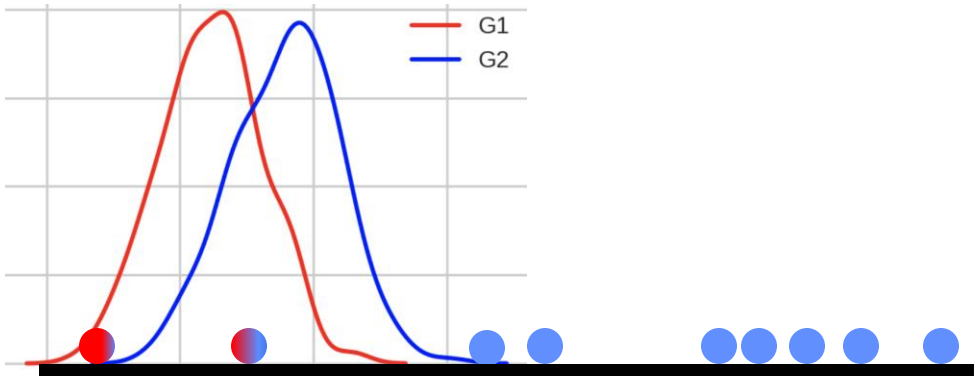
$$P(b) = \frac{b_1 + b_2 + \cdots + b_N}{N} = \frac{\sum_{i=1}^{N} b_i}{N}$$

Class prior is normalized sum of sample posteriors

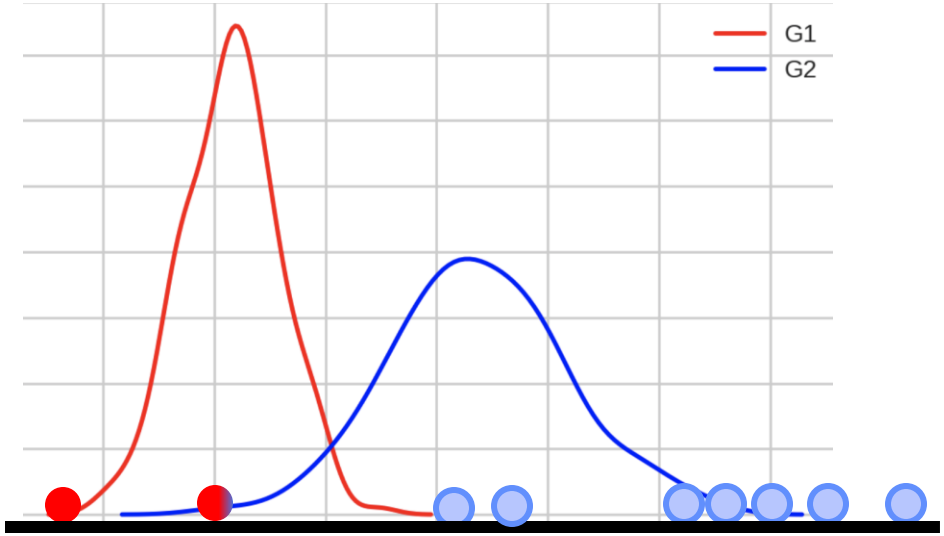
$$P(r) = \frac{\sum_{i=1}^{N} r_i}{N}$$

GMM Example: EM in action

Distributions before updating their parameters

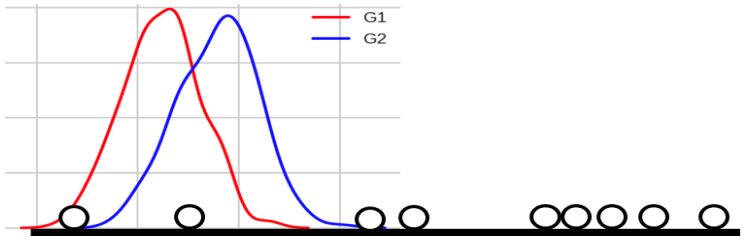


Distributions after updating their parameters using the posteriors

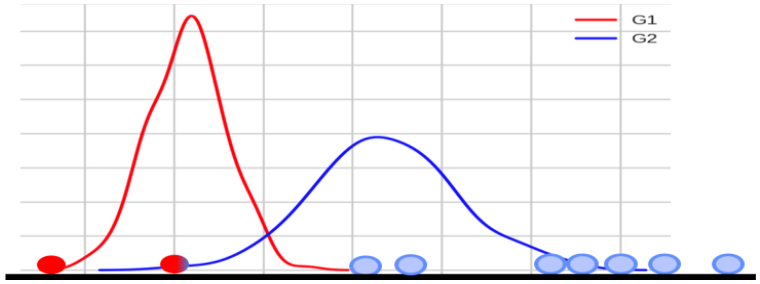


GMM Example: EM in action

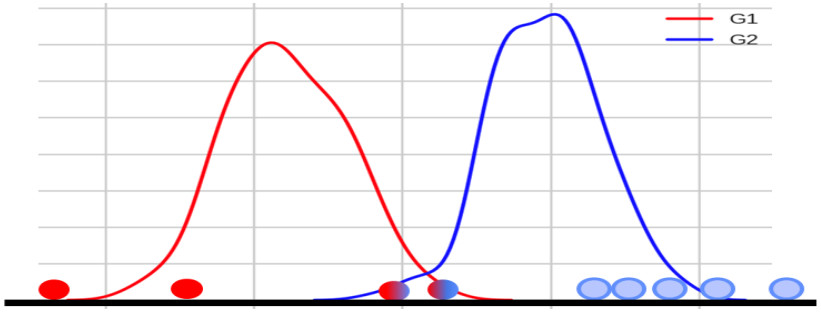
- Repeat the E and M steps iteratively till convergence
- Convergence: When M step gives the same parameters that were used in E



Initialization



1 iteration



Convergence (n iteration)

GMM: Multi-dimensional features (1)

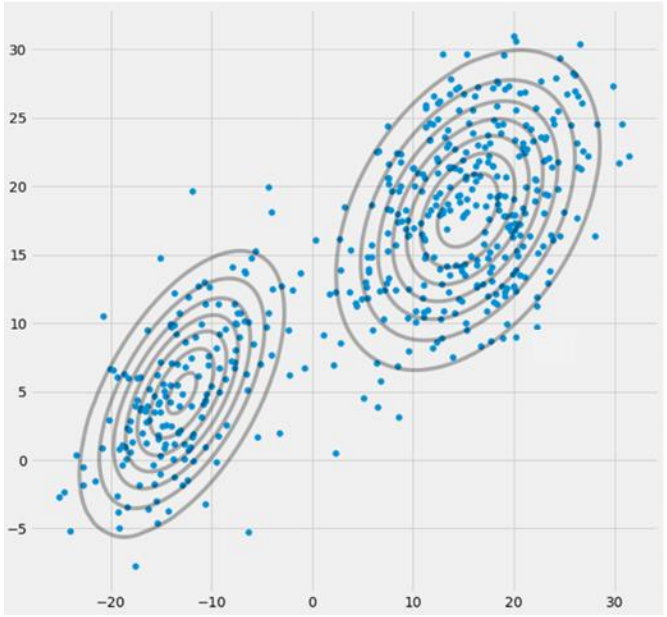
- Data with d features i.e., $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N} \in \mathbb{R}^d$ from K sources
- Each source $c \in \{1, \ldots, K\}$ has a Gaussian distribution, i.e., $\mathcal{N}(\mu_c, \Sigma_c)$ where $\mu_c \in \mathbb{R}^d$ and $\Sigma_c \in \mathbb{R}^{d \times d}$
- Iteratively estimate parameters
  - Prior: What fraction of instances came from source cluster c

$$P(c) = \frac{1}{N} \sum_{i=1}^{N} P(c|\boldsymbol{x_i})$$

  - Mean: Expected value of feature j from source cluster c:

$$\mu_{c,j} = \sum_{i=1}^{N} \left(\frac{P(c|\boldsymbol{x_i})}{N\ P(c)} \right) x_{i,j}$$

    - Similar to 1D case, but with extra index j to access specific feature from input vector



Source: https://www.python-course.eu/expectation_maximization_and_gaussian_mixture_models.php

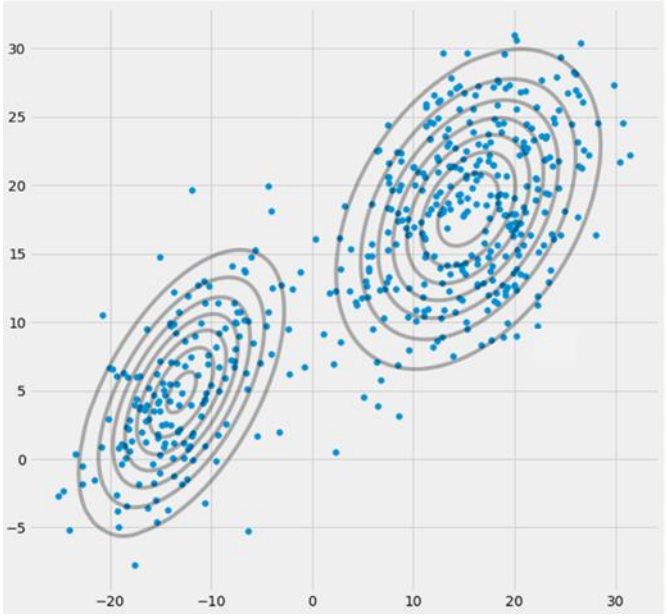
GMM: Multi-dimensional features (2)

- Data with d features i.e., $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N} \in \mathbb{R}^d$ from K sources
- Each source $c \in \{1, \ldots, K\}$ has a Gaussian distribution, i.e., $\mathcal{N}(\mu_c, \Sigma_c)$ where $\mu_c \in \mathbb{R}^d$ and $\Sigma_c \in \mathbb{R}^{d \times d}$
- Iteratively estimate parameters
  - Covariance: How related are features j and k in source c:

$$(\Sigma_c)_{j,k} = \sum_{i=1}^{N} \left(\frac{P(c|\boldsymbol{x_i})}{NP(c)} \right) (x_{i,j} - \mu_{c,j})(x_{i,k} - \mu_{c,k})$$

  - Assignment: Based on our guess of the source for each instance

$$P(c|\boldsymbol{x_i}) = \frac{p(\boldsymbol{x_i}|c)P(c)}{\sum_{c'=1}^{K} p(\boldsymbol{x_i}|c')P(c')}$$



Source: https://www.python-course.eu/expectation_maximization_and_gaussian_mixture_models.php

Picking *K* - Gaussian Components

- Maximize the log likelihood of the data given the model

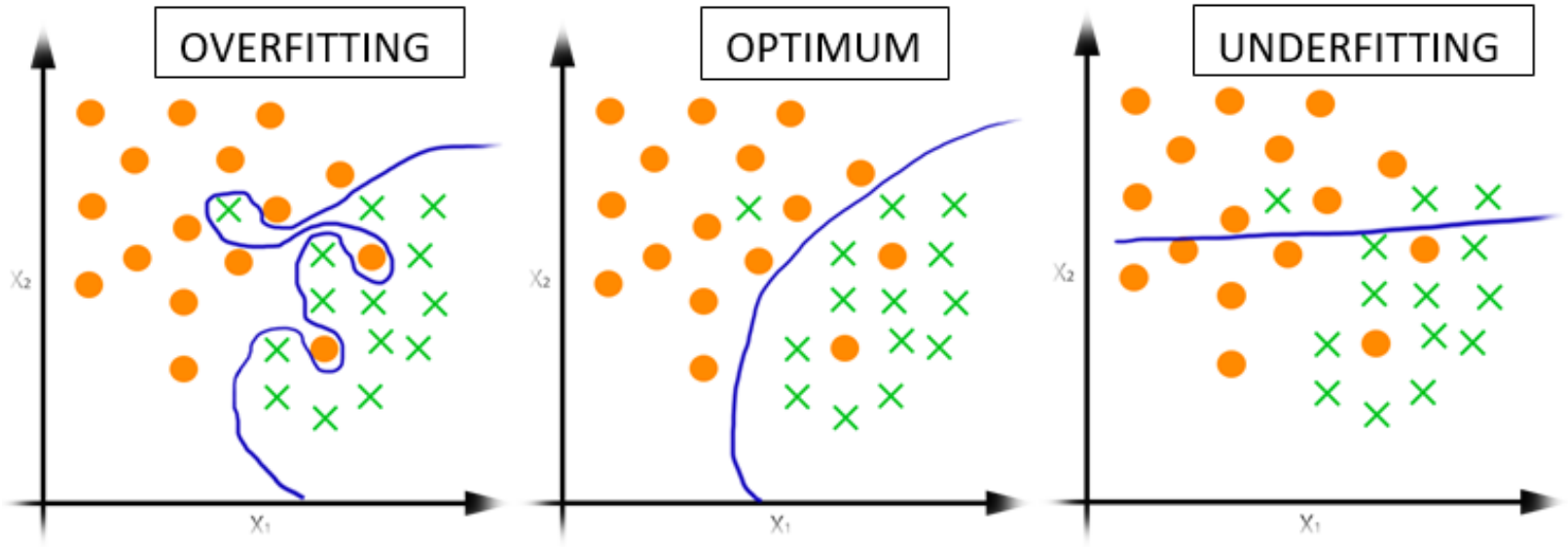
$$L = \log P(x_i, .., x_n) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} p(x_i|k)P(k)$$

- Pick *K* that makes *L* as large as possible

$$K^* = \underset{k \in \{1,..,K\}}{\operatorname{argmax}} L$$

  - *K* = *N*: each data point has its own source => overfitting
    - Unlikely to yield meaningful results for new (previously unseen) data points
    - Need to constrain (or regularize) to avoid overfitting

Overfitting



Source: https://medium.com/@srjoglekar246/overfitting-and-human-behavior-5186df1e7d19

Picking *K* - Gaussian Components

Possible to deal with overfitting using the following two ways:

- Split points into training set *T* and validation set *V*
  - For each *K,* fit parameters on *T* and measure likelihood of *V*

- **Occam's Razor**: Pick "simplest" of all models that fit
  - Bayes Inference Criterion (BIC):
    - $(\log(N)\,K\, - 2\log L)$, where K is clusters, L: log likelihood [Fraley et. al , 2002]
    - When picking from several models, the one with the lowest BIC is preferred
    - BIC introduces a penalty term for adding parameters (i.e., #clusters)
- Cross Validation

Comparing *K*-Means and GMM

Similarity between GMM and K-means

- GMM
  - Given K
1. Randomly place K Gaussians distributions
2. Calculate posterior probability for each data point for each Gaussian (soft clustering)
3. Recompute mean and variance parameters of Gaussian distributions
4. Repeat 2 & 3 until convergence

- K-means algorithm
  - Given K
1. Randomly choose K data points (seeds) to be the initial centroids i.e. cluster centers
2. Assign each data point to the closest centroid (hard clustering)
3. Recompute the centroids using the current cluster memberships
4. Repeat 2 & 3 until convergence

Calculating centroid (mean) in k-means

Say you have two clusters (K=2) and six data points ($x_1, x_2, \ldots x_6$). Assume that (x_1, x_4, x_5) belong to cluster 'a' and (x_2, x_3, x_6) belong to cluster 'b'

For k-means, centroid of cluster a:

$$centroid_a = \frac{x_2 + x_4 + x_5}{3} = \frac{x_1(0) + x_2(1) + x_3(0) + x_4(1) + x_5(1) + x_6(0)}{1(0) + 1(1) + 1(0) + 1(1) + 1(1) + 1(0)}$$

- In the rightmost expression, x_i is multiplied with 1 if x_i belongs to cluster a and 0 if it does not.

Calculating mean in GMM

- If we were doing GMM, then the mean of cluster a (μ_a) is

$$\mu_a = \frac{x_1 P(a|x_1) + x_2 P(a|x_2) + x_3 P(a|x_3) + \ldots + x_6 P(a|x_6)}{1(P(a|x_1)) + 1(P(a|x_2)) + 1(P(a|x_3)) + \ldots + 1(P(a|x_6))}$$

Comparing formulae for means

- Notice the similarity between

$$\frac{x_1\,(0) + x_2(1) + x_3(0) + x_4(1) + x_5(1) + x_6(0)}{1(0) + 1(1) + 1(0) + 1(1) + 1(1) + 1(0)}$$

And

$$\frac{x_1 P(a|x_1) + x_2 P(a|x_2) + x_3 P(a|x_3) + \ldots ++ x_6 P(a|x_6)}{1(P(a|x_1)) + 1(P(a|x_2)) + 1(P(a|x_3)) + \ldots ++ 1(P(a|x_6))}$$

- Calculation of the mean involves:
  - Multiplying by 0 or 1 in k-means (hard clustering)
  - Multiplying by posterior probability (between 0 and 1) in GMM (soft clustering)

Summary

- K-means is a hard-clustering whereas GMMs is a soft-clustering method

- GMMs and K-means: Similarity
  - Sensitive to starting point, converges to local maximum
  - Convergence: When change in $P(x_1, x_2, \ldots, x_n)$ is sufficiently small
  - Cannot discover k easily

- Can make GMMs to behave as K-means
  - Fix variance to be 1
  - Uniform priors

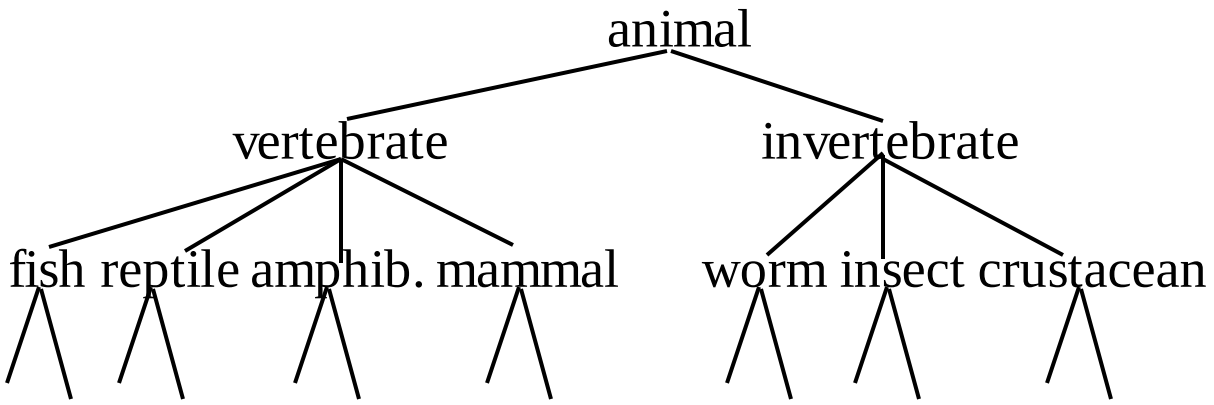
Hierarchical Clustering

Ways to do clustering

- Agglomerative vs Divisive
  - *Agglomerative*: each instance is its own cluster and the algorithm merges clusters
  - *Divisive*: begins with all instances in one cluster and the algorithm divides it up
- Hard vs Soft/Fuzzy
  - Hard clustering assigns each instance to one cluster
  - Soft/Fuzzy clustering assigns degree of membership

Hierarchical Clustering

- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of documents.



- *One approach*: recursive application of a partitional clustering algorithm.

Hierarchical Clustering

- Agglomerative approach

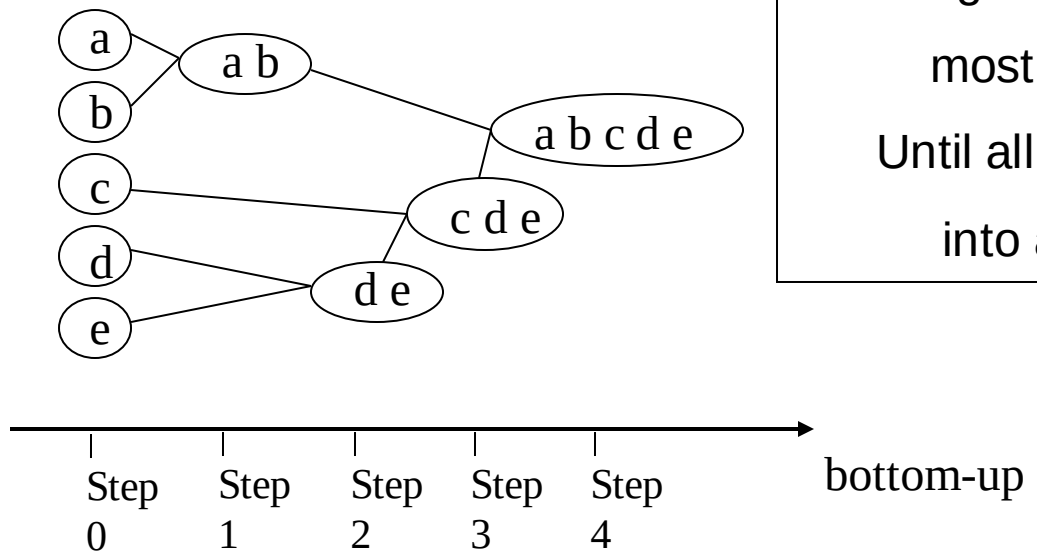
Initialization:

Each object is a cluster

Iteration:

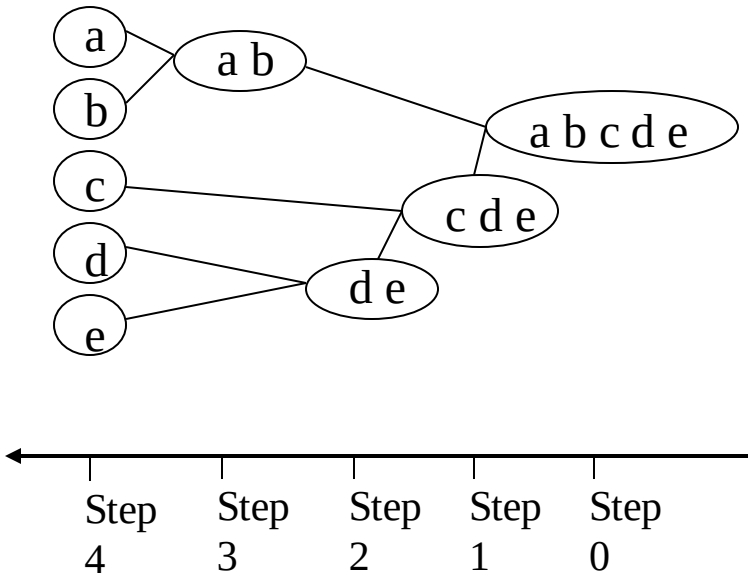
Merge two clusters which are most similar to each other;

Until all objects are merged into a single cluster



Hierarchical Clustering

- Divisive Approaches



Initialization:

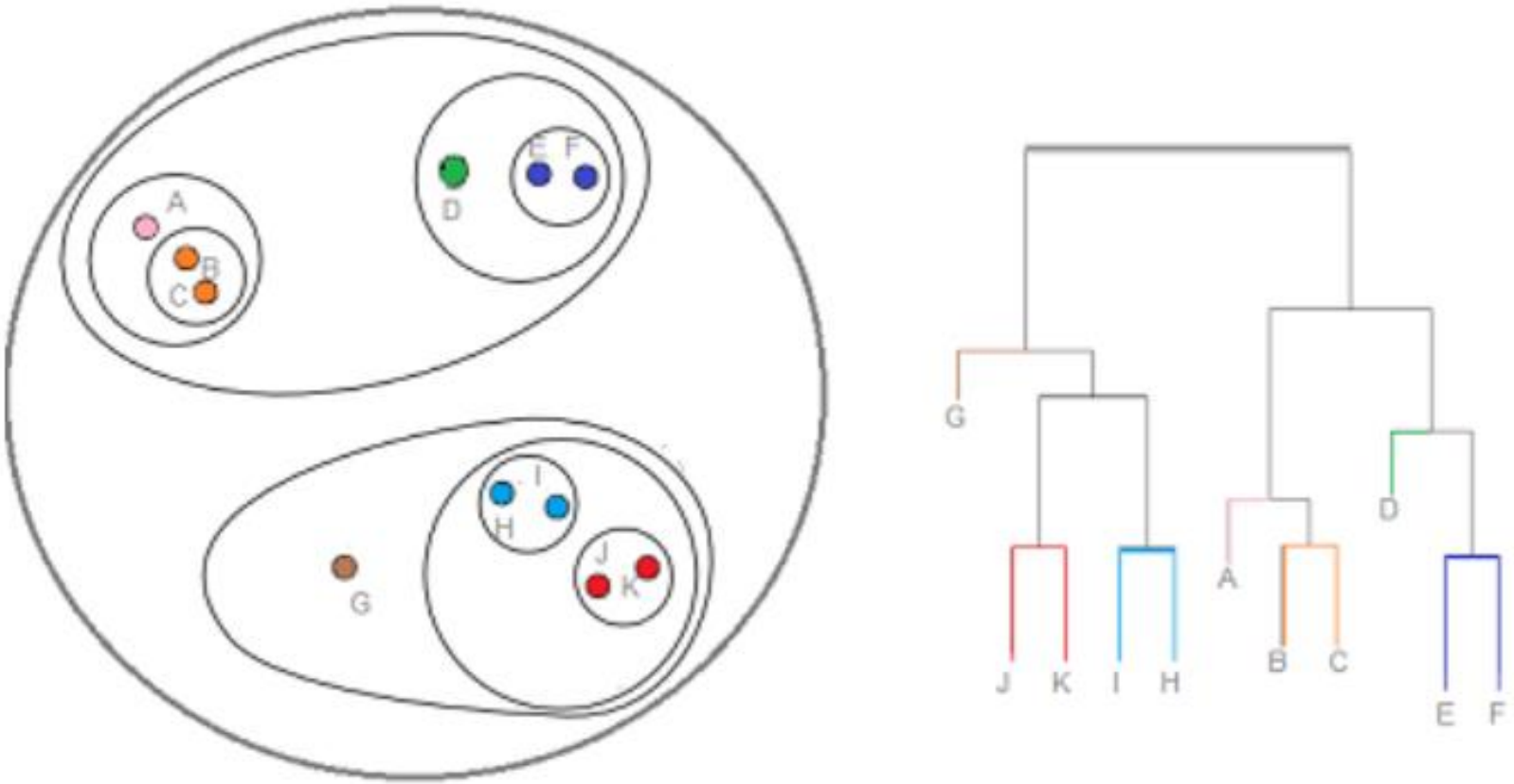
All objects stay in one cluster

Iteration:

Select a cluster and split it into two sub clusters

Until each leaf cluster contains only one object

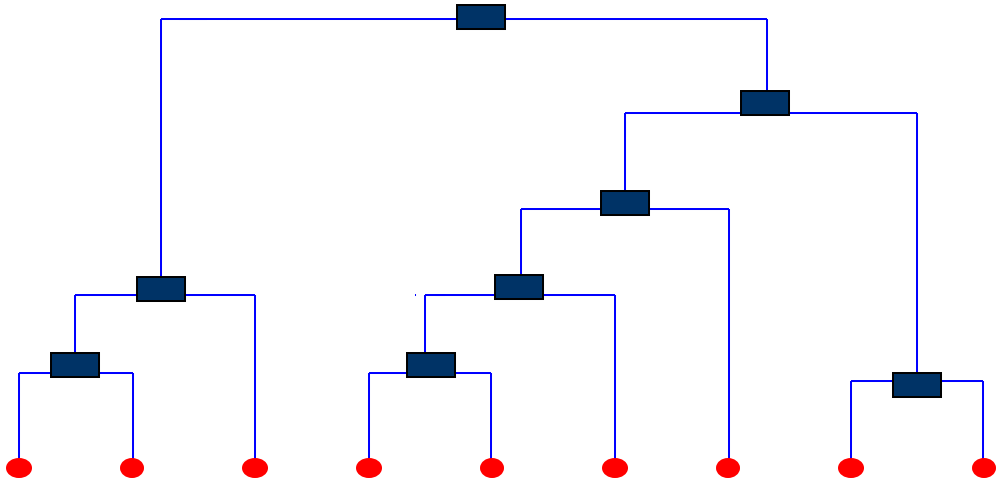
Dendrogram



A dendrogram represents nested clusters

Dendrogram

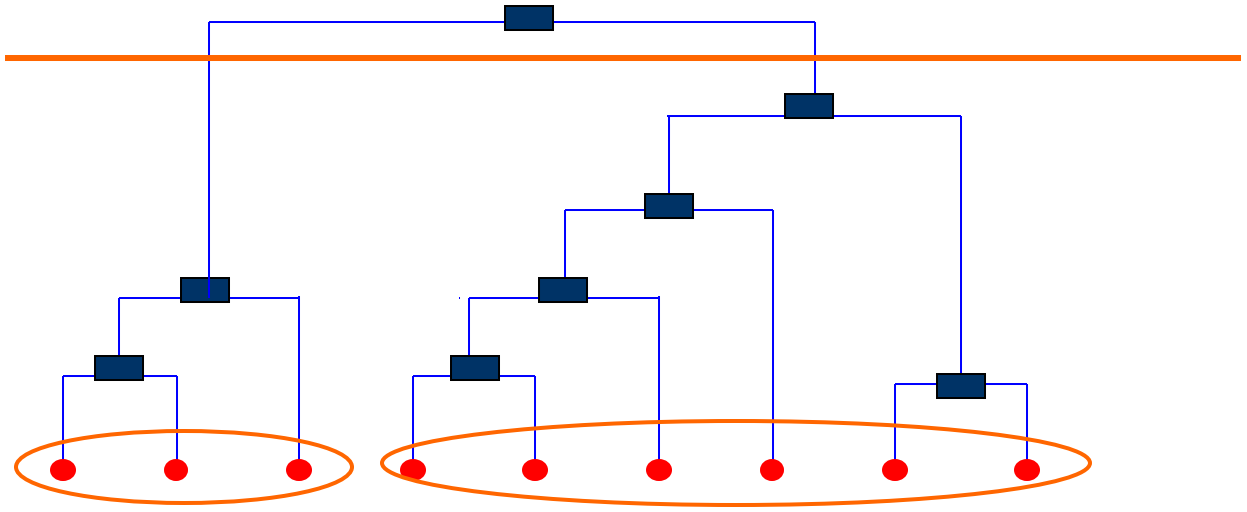
- A binary tree that shows how clusters are merged/split hierarchically
- Each node on the tree is a cluster; each leaf node is a singleton cluster



Example: How points are clustered

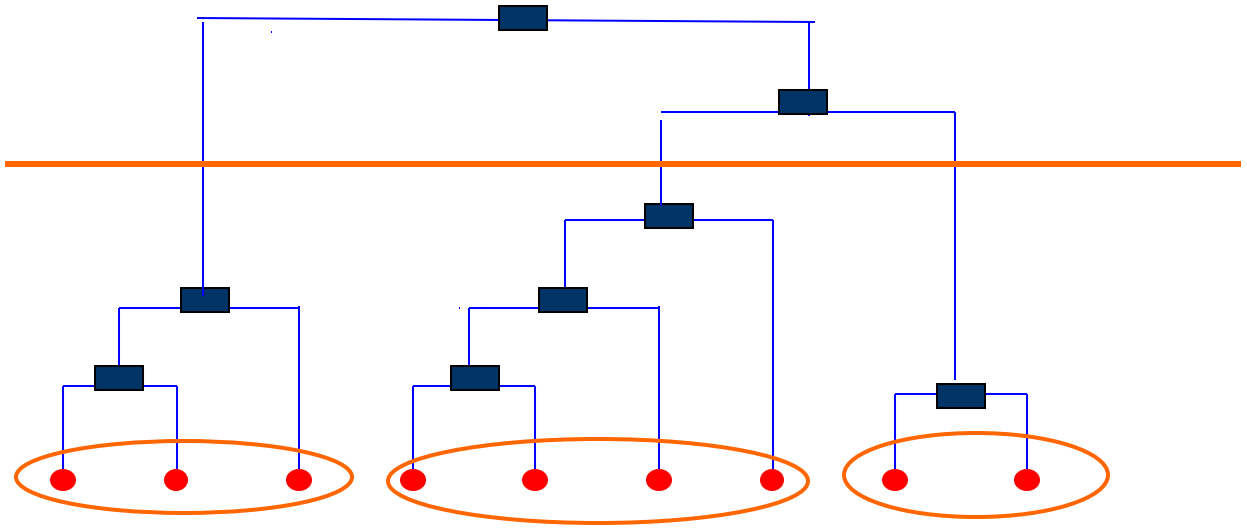
Dendrogram

- A clustering of the data objects is obtained by cutting the *dendrogram* at the desired level, then each connected component forms a cluster



Dendrogram

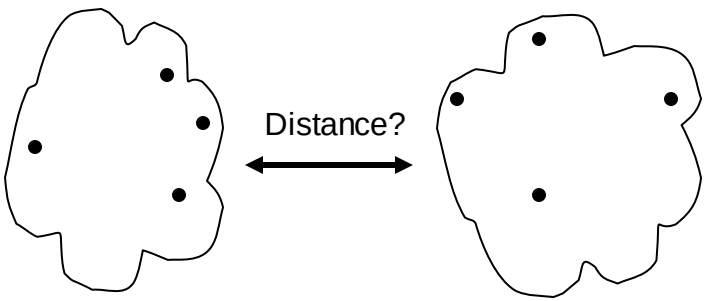
- A clustering of the data objects is obtained by cutting the *dendrogram* at the desired level, then each connected component forms a cluster



How to Merge Clusters?

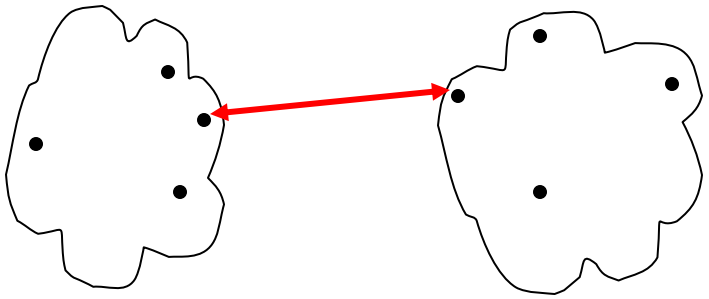
- How to measure the distance between clusters?

Single-link
Complete-link
Average-link
Centroid distance



Hint: *Distance between clusters* is usually defined on the basis of *distance between objects.*

How to Define Inter-Cluster Distance



Single-link

Complete-link

Average-link

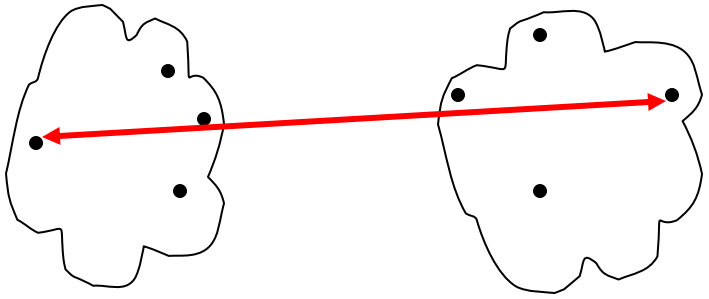
Centroid distance

$$d_{min}(C_i, C_j) = \min_{p \in C_i, q \in C_j} d(p, q)$$

- The distance between two clusters is represented by the distance of the *closest pair of data objects* belonging to different clusters.

- Can result in "straggly" (long and thin) clusters due to chaining effect

How to Define Inter-Cluster Distance



Single-link

Complete-link

Average-link

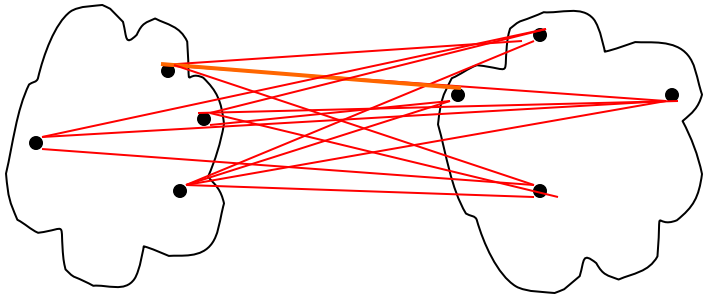
Centroid distance

$$d_{max}(C_i, C_j) = \max_{p \in C_i, q \in C_j} d(p, q)$$

- The distance between two clusters is represented by the distance of the *farthest pair of data objects* belonging to different clusters.

- Makes tighter spherical clusters that are typically preferred

How to Define Inter-Cluster Distance



Single-link

Complete-link

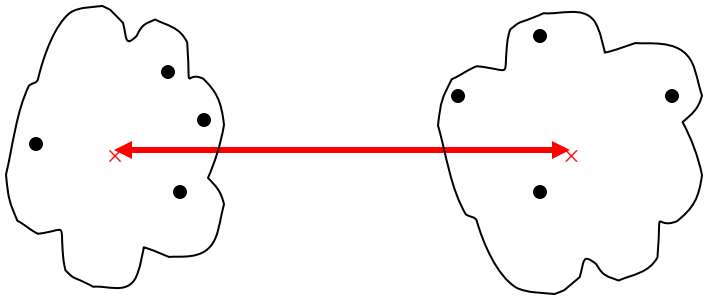
Average-link

Centroid distance

$$d_{avg}(C_i, C_j) = \underset{p \in C_i, q \in C_j}{\text{avg}} \; d(p, q)$$

The distance between two clusters is represented by the *average* distance of *all pairs of data objects* belonging to different clusters.

How to Define Inter-Cluster Distance



m_i, m_j are the means of C_i, C_j,

Single-link

Complete-link

Average-link

Centroid distance

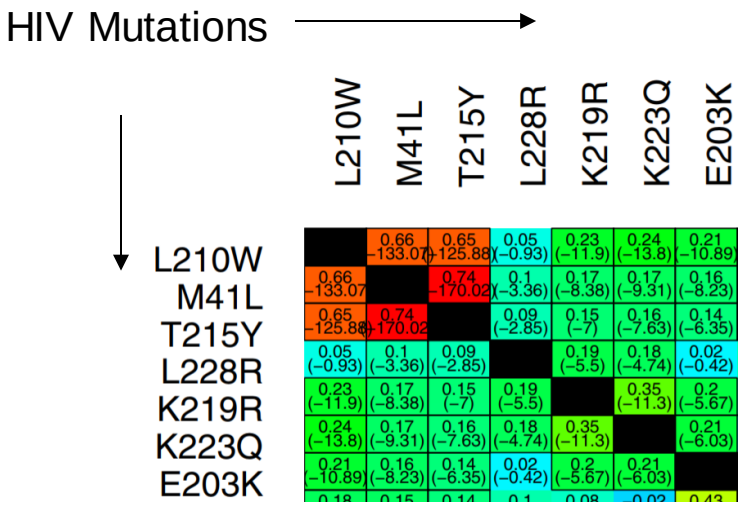
$$d_{mean}(C_i, C_j) = d(m_i, m_j)$$

The distance between two clusters is represented by the distance between *the means of the clusters.*

Hierarchical Clustering Example

Characterization Novel HIV Drug Resistance Mutations using Clustering.

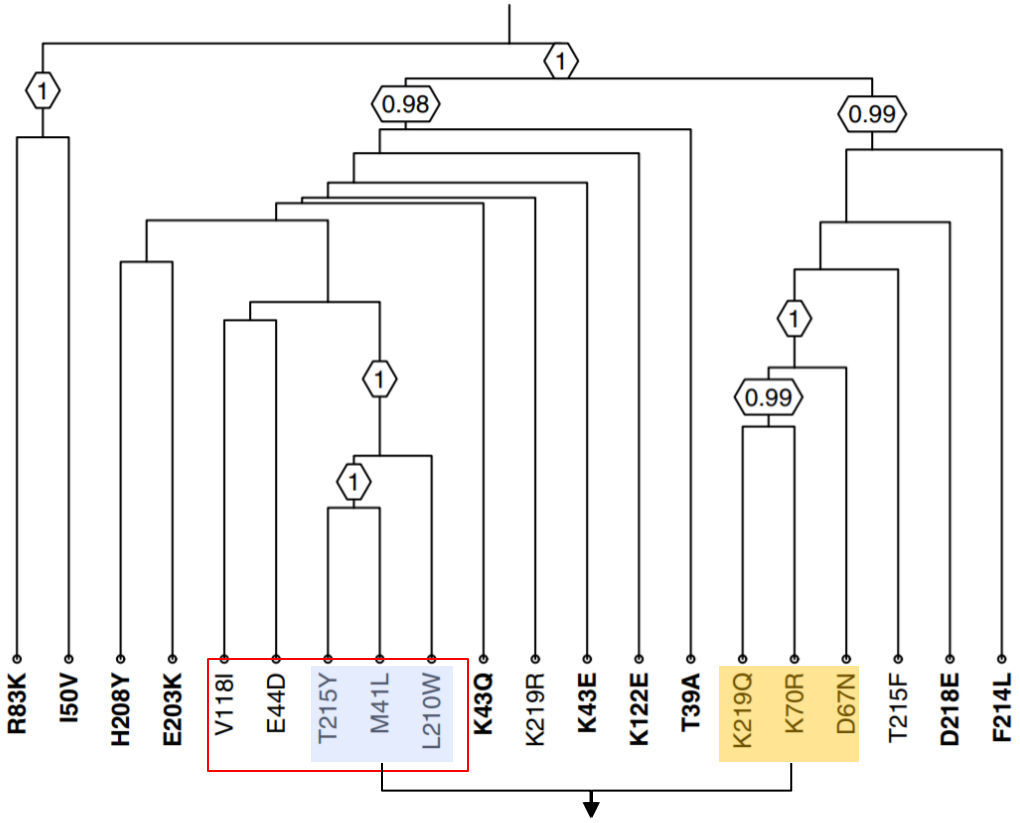
- **Objective:** By clustering new HIV mutations with HIV mutations that have known drug resistance mechanisms, we can infer the possible drug resistance mechanisms of the new mutations
- **Clustering Technique:** Agglomerative Hierarchical Clustering using Average-link
- **Distance Metric:** Matthews correlation coefficient
  - This coefficient measures how two individual mutations vary together in the population.



Reference: Sing, Tobias, et al. "Characterization of novel HIV drug resistance mutations using clustering, multidimensional scaling and SVM-based feature ranking." *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, Berlin, Heidelberg, 2005

Hierarchical Clustering Example

- **Dendrogram after clustering**

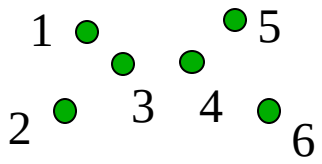


Mutations with known drug resistance mechanisms

- Clustering is performed on HIV mutations with known and unknown drug resistance mechanisms
- Mutation complexes in shaded boxes have known drug resistance mechanisms
- Mutations with unknown drug resistance mechanisms may have similar drug resistance mechanisms with other mutations they are clustered with.
- E.g, Mutations E44D and V118I may have similar resistance mechanisms to the blue mutation complex, for which a drug resistance mechanism is already known.

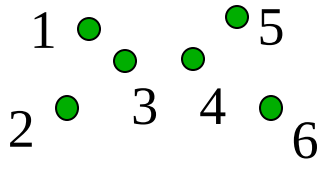
An Example of the Agglomerative Hierarchical Clustering Algorithm

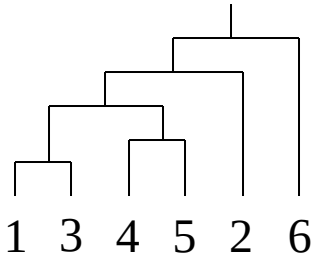
- For the following data set, we will get different clustering results with the single-link and complete-link algorithms.



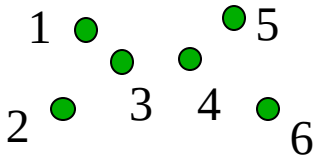
Results

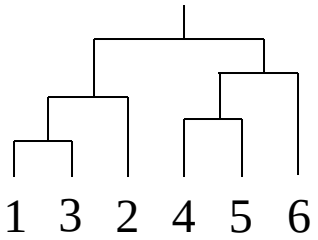
Single Link algorithm





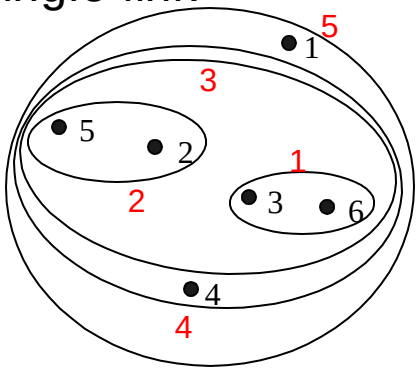
Complete Link algorithm



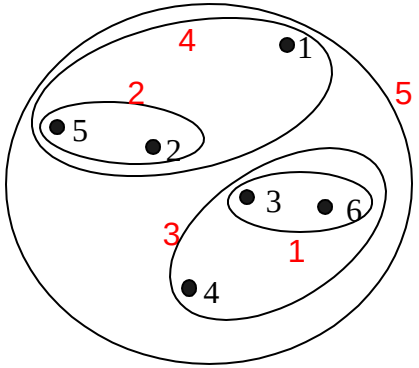


Hierarchical Clustering: Comparison

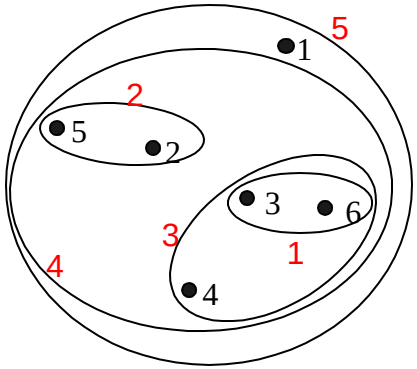
Single-link



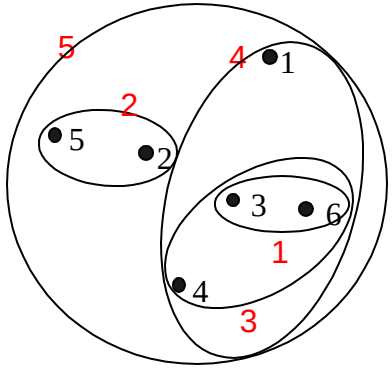
Complete-link



Average-link

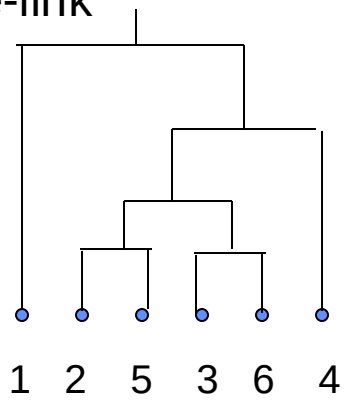


Centroid distance

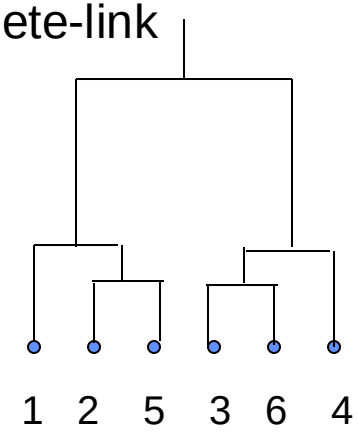


Compare Dendrograms

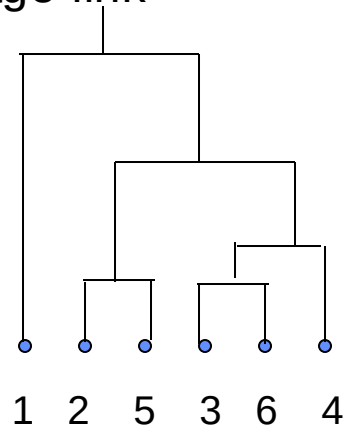
Single-link



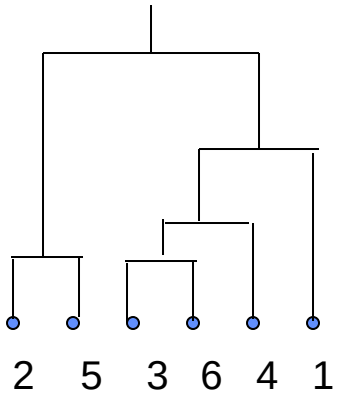
Complete-link



Average-link

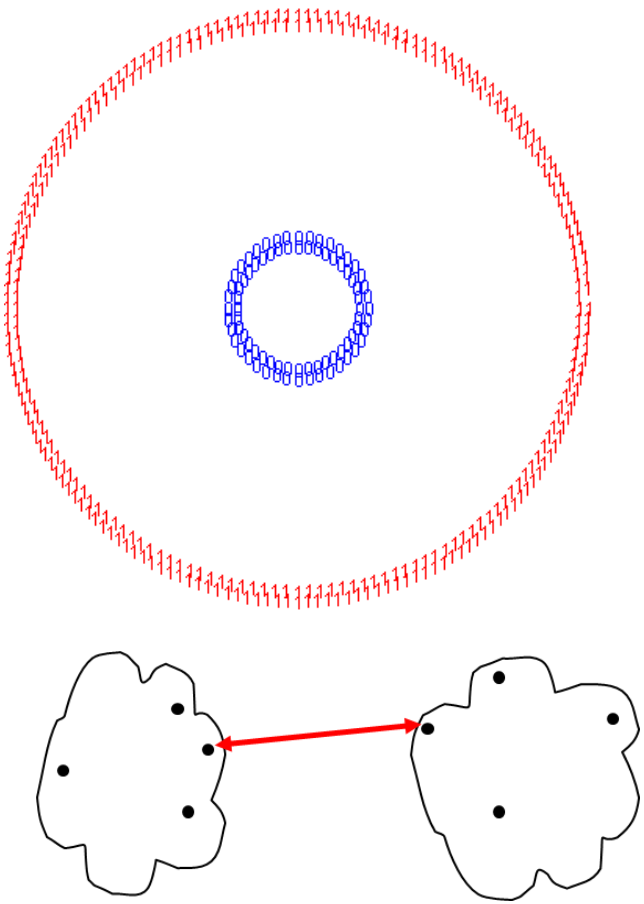


Centroid distance

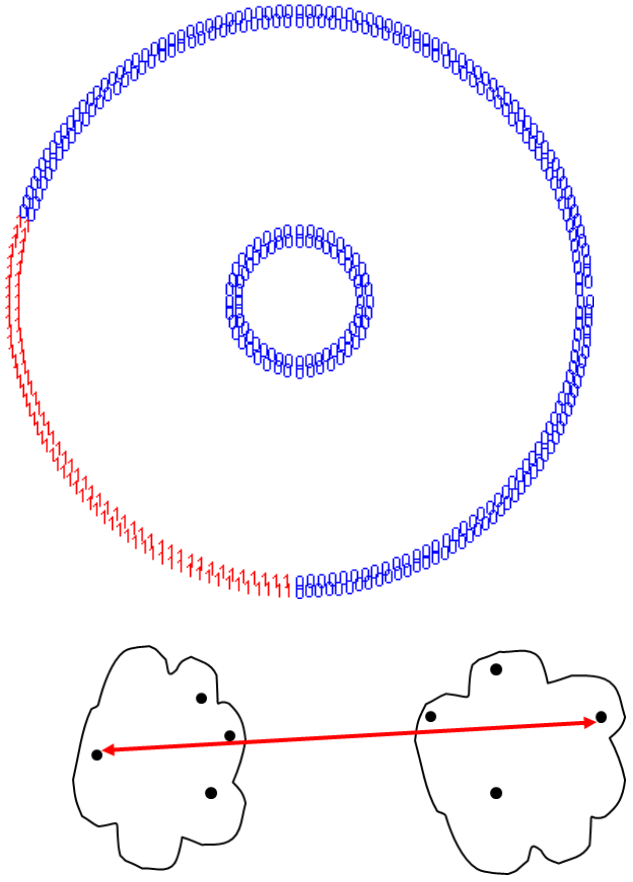


Effect of Bias towards Spherical Clusters

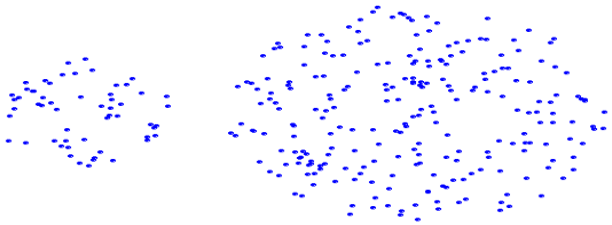
Single-link (2 clusters)



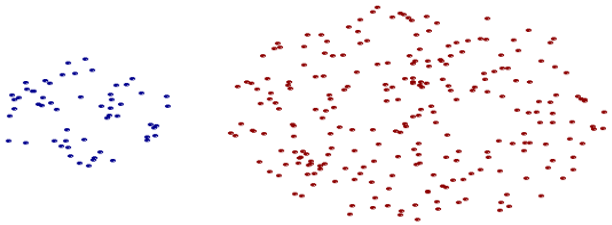
Complete-link (2 clusters)



Strength of Single-link



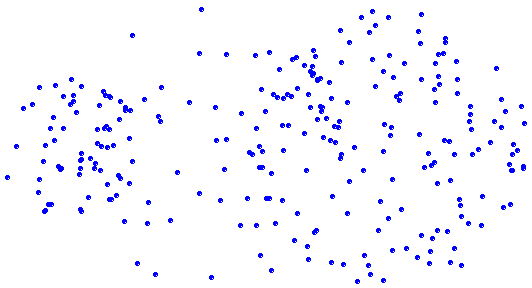
Original Points



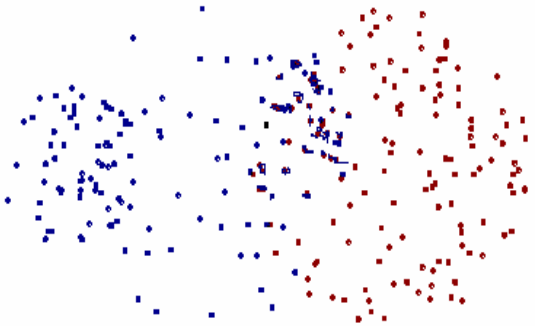
Two Clusters

- Can find irregular cluster shapes

Limitations of Single-Link



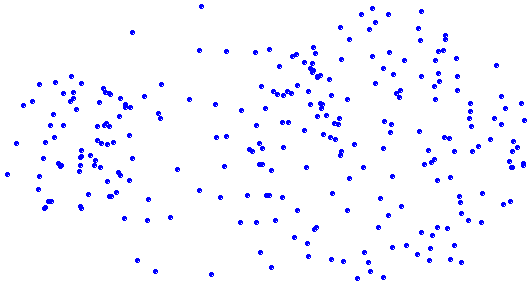
Original Points



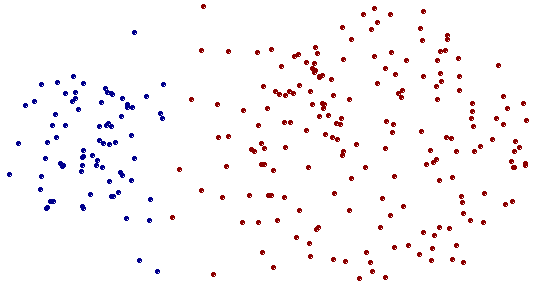
Two Clusters

- Sensitive to noise and outliers

Strength of Complete-link



Original Points



Two Clusters

- Less susceptible to noise and outliers

Which Method is Better?

- Each method has its own advantages and disadvantages; application-dependent, single-link and complete-link are the most common methods
- Single-link
  - Can find irregular-shaped clusters
  - Sensitive to outliers, suffers the so-called chaining effect
- Complete-link, Average-link, and Centroid distance
  - Robust to outliers
  - Tend to break large clusters
  - Prefer spherical clusters

Another similarity measure

- In the examples described above, we used Euclidean distance to find the distance between points/clusters
- Depending on the type of the data, other similarity measures (measures of distance) might be preferred such as **correlation-based distance**
- Correlation-based distance considers two observations to be similar if their features are highly correlated, even though the observed values may be far apart in terms of Euclidean distance
- If Euclidean distance is chosen, then observations with high values of features will be clustered together. The same holds true for observations with low values of features.