DATA MINING

Cluster Analysis

Hierarchical, Density, Model Clustering

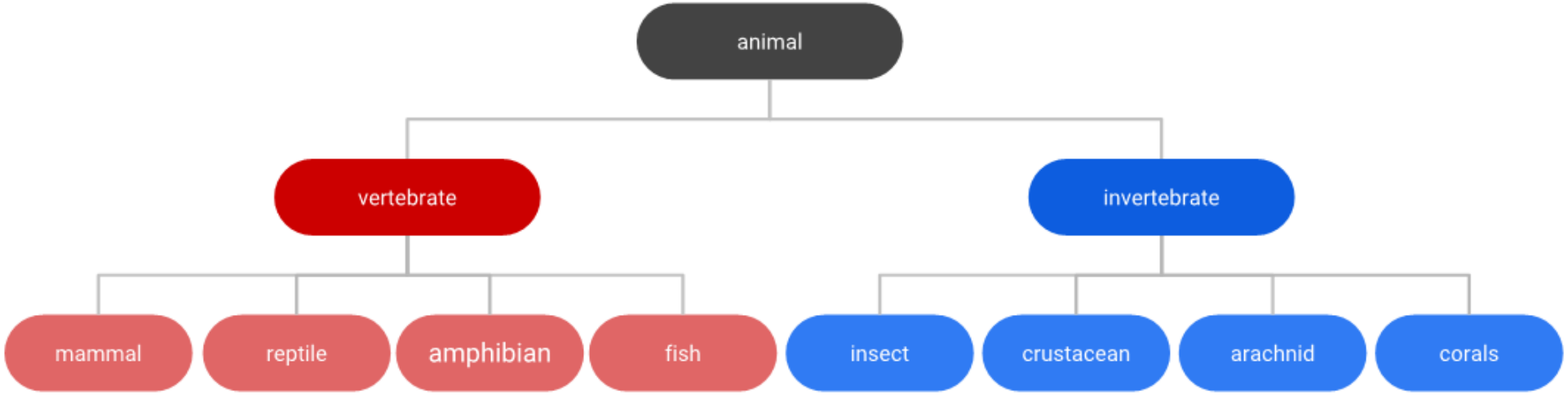
Ashish Pujari

Lecture Outline

- Hierarchical Clustering
  - Agglomerative and Divisive clustering
- Density Based Clustering
  - DBSCAN
- Model Based Clustering
  - GMM

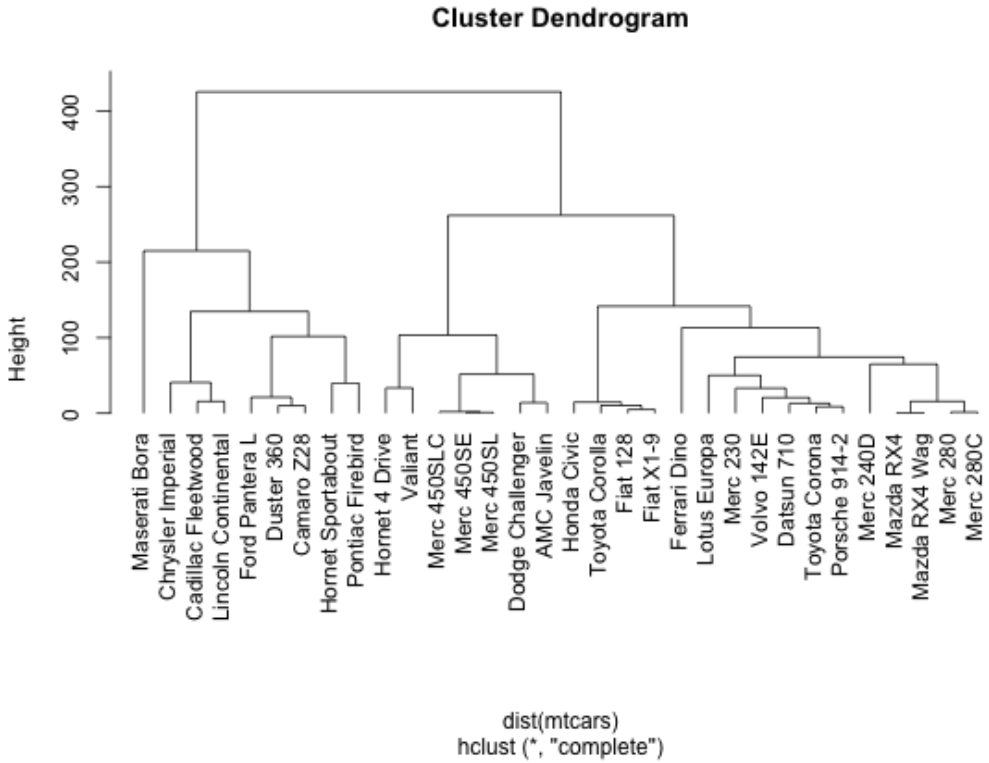
Hierarchical Clustering

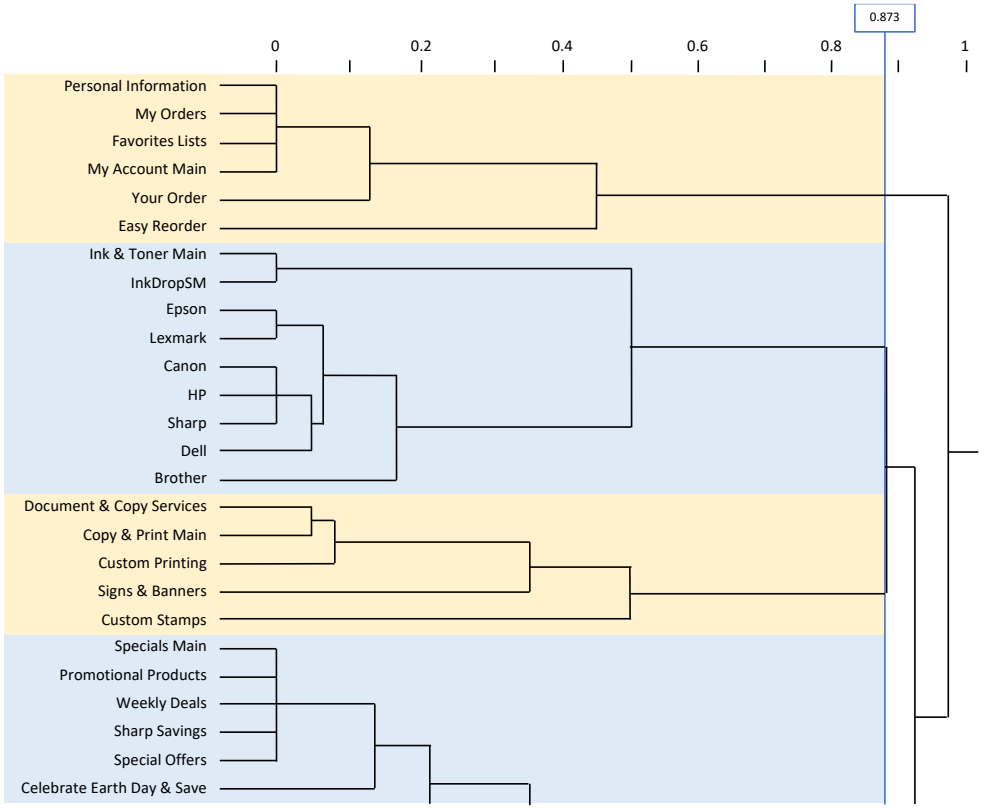
- Hierarchical (or connectivity based) clustering creates groups so that objects within a group are similar to each other and different from objects in other groups
- Approach
  - Core Idea - Objects are more related to nearby objects than to objects farther away
  - Does not only provide an answer for every *k*, but shows the data at different levels of granularity



Dendrogram

- Cluster hierarchy (multi-level nested partitioning) is displayed as a dendrogram
- Objects are placed along the x-axis such that the clusters do not mix
- y-axis marks the distance at which the clusters merge

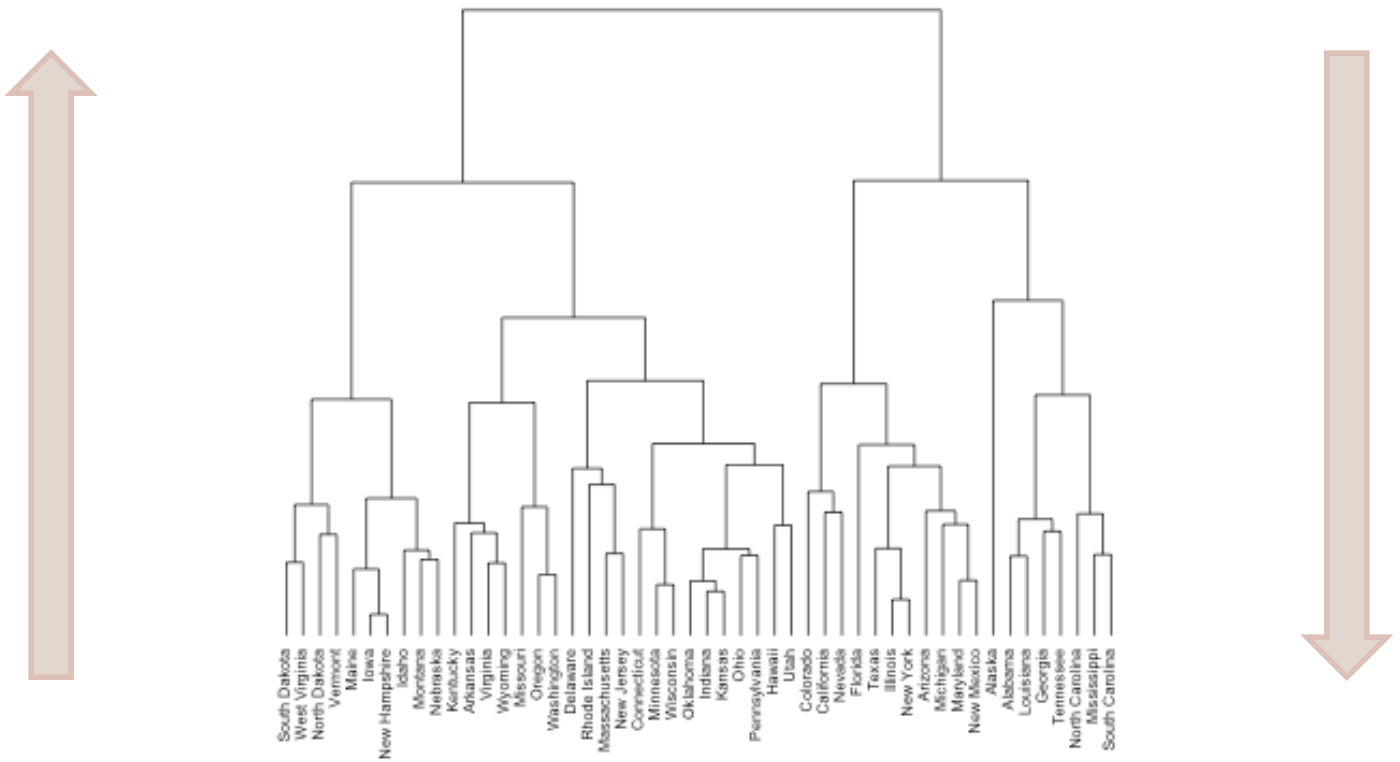




Hierarchical Clustering: Approaches

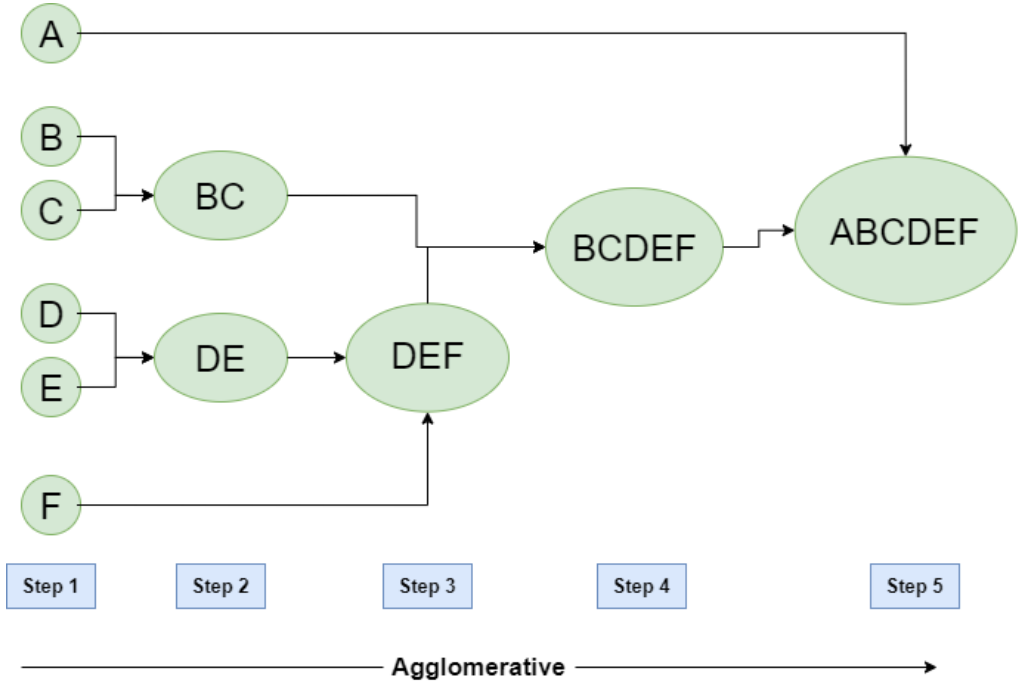
Agglomerative (Bottom-up):
- Start with individual clusters
- At each step, merge the most similar or closest pair of cluster

Divisive (Top-down):
- Start with a single cluster (universe)
- At each step, split clusters until only singleton clusters remain

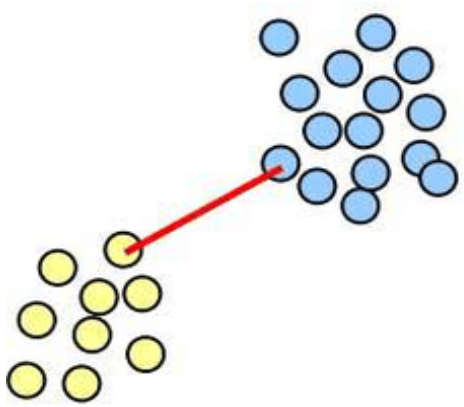


Agglomerative Clustering

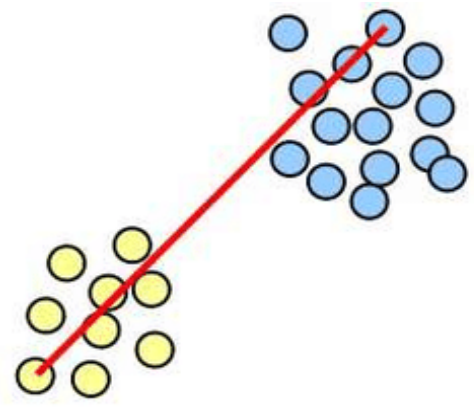
- Approach
  - Initially each data point is considered as an individual cluster.
  - At each iteration, the similar clusters merge with other clusters until one cluster or K clusters are formed.
- Algorithm
  - Let each data point be a cluster
  - **Repeat**:
    - Compute the pairwise proximity matrix (ij^{th} entry gives the similarity between the i^{th} and j^{th} clusters)
    - Merge the two most similar clusters and update the proximity matrix
  - **Until** only a single cluster remains



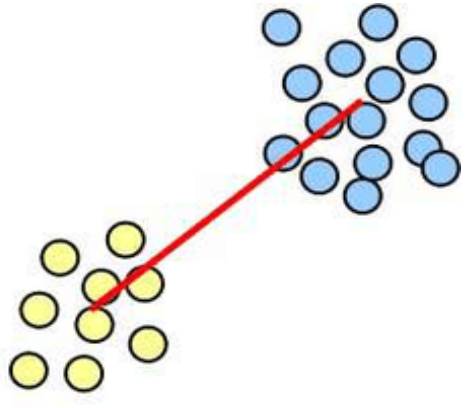
Cluster Distance Metrics



single-link



complete-link



average-link

Single Link	Complete Link	Average Link
Similarity between two clusters is the similarity between their most similar (nearest neighbor) members	The similarity between two clusters is the similarity between their most dissimilar members	The similarity between two clusters is the average distance between members
Local similarity-based: Emphasizing more on close regions, ignoring the overall structure of the cluster	Nonlocal in behavior, the entire structure of the clustering can influence merge decisions resulting in compact clusters (smallest diameter)	Average dissimilarity between all the pairwise dissimilarities
Sensitive to noise and outliers	Sensitive to outliers	Not very sensitive to outliers

Ward's Method

- Ward's Criterion
  - Greedy algorithm for the hierarchical k-means problem which ends to produce more compact clusters
  - Criterion for choosing the pair of clusters to merge at each step is based on an objective function.
  - Minimizes the total within-cluster variance and find the pair of clusters that leads to minimum increase in total within-cluster variance after merging

$$\Delta(A,B) = \sum_{i \in AUB} \|x_i - m_{AUB}\|^2 - \sum_{i \in A} \|x_i - m_{\mathrm{A}}\|^2 - \sum_{i \in B} \|x_i - m_B\|^2 = \frac{n_A n_B}{n_A + n_B} \|m_A - m_B\|^2$$

where Δ the merging cost of combining the clusters A and B

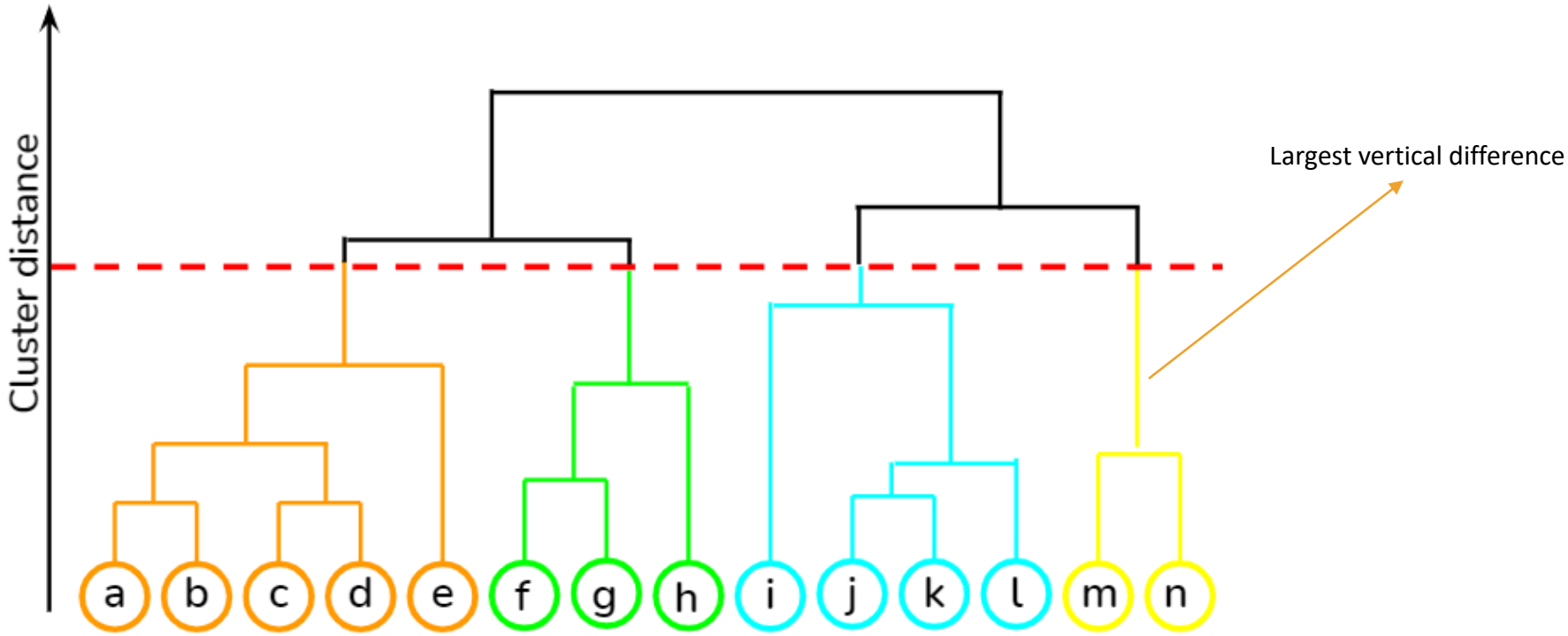
n_A, n_B number of points in clusters A and B

$m_{\mathrm{A}}, m_{\mathrm{B}}$ center of clusters A and B

- Further Reading
  - Analysis of Ward's Method
  - Generalizing Ward's Method for Use with Manhattan Distances

Optimal Number of Clusters

- Locate the largest vertical difference between nodes and cut using a horizontal line. The number of vertical lines intersecting it is the optimal number of clusters



Metrics

- Pseudo-F

$$\text{Pseudo} - \text{F} = \frac{\text{variance between clusters}}{\text{variance within clusters}} = \frac{SS\ between\ clusters/\ (k-1)}{SS\ within\ clusters/\ (n-k)}$$

  - The higher the number the better

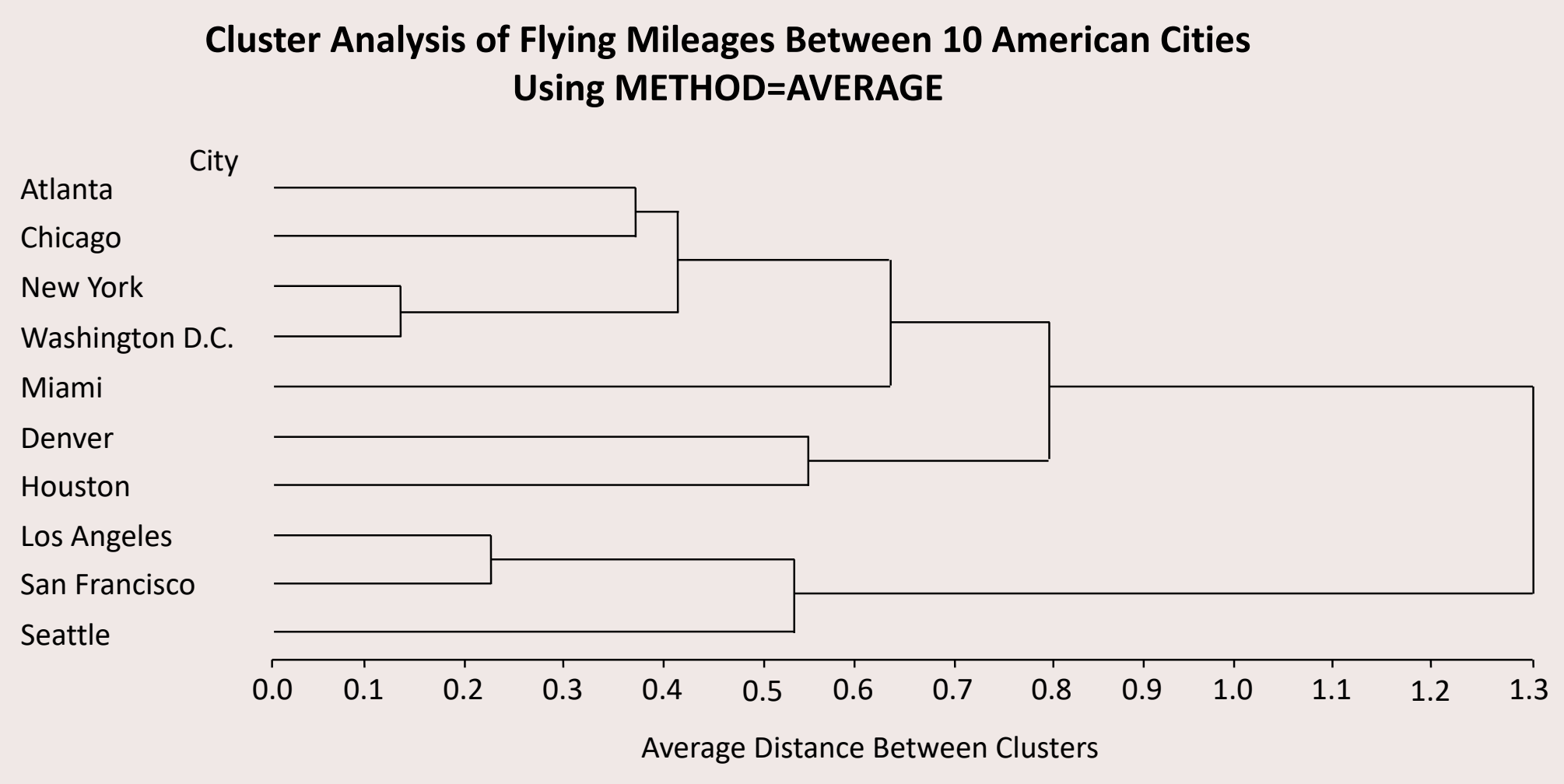
- Pseudo T-square
  - Clusters p, q merged
  - Difference between two clusters
  - Stop when distinct jump

$$Pseudo\,T-square = \frac{between\ cluster\ SS\,(p,q)}{(within\ SS\,(p) + within\ SS\,(q))\ /\ (n_p + n_q - 2)}$$

Example: Flying Mileages

ATL	ORD	DEN	IAD	LAX	MIA	LGA	SFO	SEA	DCA	
0										Atlanta
587	0									Chicago
1212	920	0								Denver
701	940	879	0							Houston
1936	1745	831	1374	0						Los Angeles
604	1188	1726	968	2339	0					Miami
748	713	1631	1420	2451	1092	0				New York
2139	1858	949	1645	347	2594	2571	0			San Francisco
2182	1737	1021	1891	959	2734	2408	678	0		Seattle
543	597	1494	1220	2300	923	205	2442	2329	0	Washington D.C.

Example: Flying Mileages

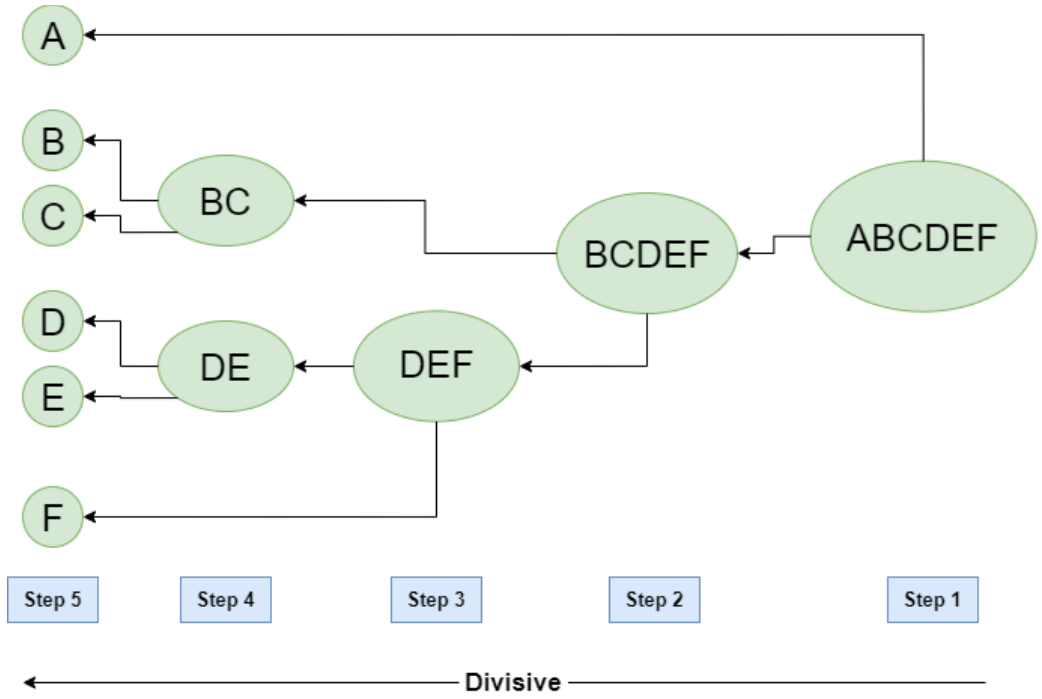


Example: Flying Mileages

NCL	Clusters	Joined	FREQ	PSF	PST2	Norm RMS Dist
9	New York	Washington D.C.	2	66.7	.	0.1297
8	Los Angeles	San Francisco	2	39.2	.	0.2196
7	Atlanta	Chicago	2	21.7	.	0.3715
6	CL7	CL9	4	14.5	3.4	0.4149
5	CL8	Seattle	3	12.4	7.3	0.5255
4	Denver	Houston	2	13.9	.	0.5562
3	CL6	Miami	5	15.5	3.8	0.6185
2	CL3	CL4	7	16.0	5.3	0.8005
1	CL2	CL5	10	.	16.0	1.2967

Divisive Clustering

- Approach
  - Top down, global approach
  - Starts off with all the points into one cluster and divides them to create more clusters.
  - Weakens the connection between objects in different clusters
- Algorithm
  - Start at the root (top) with all the points as one cluster
  - **Repeat** Splits the cluster into two using the splitting criterion
  - **Until** target number of clusters are obtained (or all clusters are singletons)



Divisive Clustering: Considerations

- Choosing which cluster to split
  - Check the sums of squared errors of the clusters and choose the one with the largest value
- Determining how to split
  - Because there exist $O(2^n)$ ways of splitting each cluster, heuristics are needed
  - Ward's criterion for greater reduction in the difference in the SSE criterion after a split
  - For categorical data, Gini-index may be used
- Noise handling
  - Use a threshold to determine the termination criterion (very small clusters can be noise)

Bisecting K-Means

- Hybrid approach between Divisive Clustering and K-means Clustering
- Approach
  - Instead of creating k clusters in each iteration, it splits one cluster into two sub clusters at each bisecting step (by using k-means, k=2) until k clusters are obtained
  - Bisection Strategies
    - Largest cluster: Selects the cluster having the most points
    - Biggest inertia: Cluster with biggest SSE within
- Pros
  - More efficient than K-means
    - When the number of clusters is large
    - Only works on a subset of the data at each bisection, while K-means always works on the entire dataset.
  - More efficient than agglomerative clustering
    - If the number of clusters is small compared to the number of data points.
- Further Reading
  - Performance Analysis of K-Means and Bisecting K-Means Algorithms in Weblog Data

Agglomerative vs Divisive

	Agglomerative	Divisive
Approach	• Bottom-up methods make clustering decisions based on local patterns	• Top-down methods benefits from complete information about global patterns
Cluster Quality	• Global decisions made later • More greedy approach • Poor solution quality	• Global decisions made early • Early decisions cannot be undone • Better solution quality
Implementation Complexity	• Simpler algorithm	• More complex as we need a second, flat clustering algorithm as a “subroutine” such as k-means
Scalability	• Does not scale well to large datasets	• Does not scale well to large datasets
Time Complexity	• $O(n^3)$ or $O(n^2)$ for more efficient variations • High due to the need to calculate and recalculate full pairwise distance matrices.	• O(n): linear in the number of patterns and clusters. • $O(2^n)$ with an exhaustive search • Common to use faster heuristics to choose splits, such as k-means

https://en.wikipedia.org/wiki/Hierarchical_clustering

K-Means vs Hierarchical

	K-means	Hierarchical
Approach	• Division of the set of data objects into non-overlapping subsets (clusters).	• A hierarchical clustering is a set of nested clusters that are arranged as a tree.
Distance Metrics	• Distance based on norm (typically Euclidian or Manhattan)	• Handles of any form of similarity or distance metric.
Cluster shapes	• Works well when the structure of the clusters is hyper spherical	• Does not form spherical clusters • Shows all possible linkages between clusters
Number of Clusters	• It requires advance knowledge of 'K'. This can be difficult to determine	• No need to preset the number of clusters • Find appropriate number by interpreting the dendrogram.
Algorithms	• Standard K-means, K-means++ • K-medoids, K-medians	• Ward method • Single, Complete, Average Linkage, etc. • DIANA (Divisive Analysis) , AGNES
Scalability	• Scales well for small to medium sized datasets • Can be made to scale with modifications	• Scales well for smaller datasets • Not easy to visualize dendrograms for large datasets • Requires computation and storage of an $n \times n$ distance matrix. For very large datasets, this can be expensive and slow

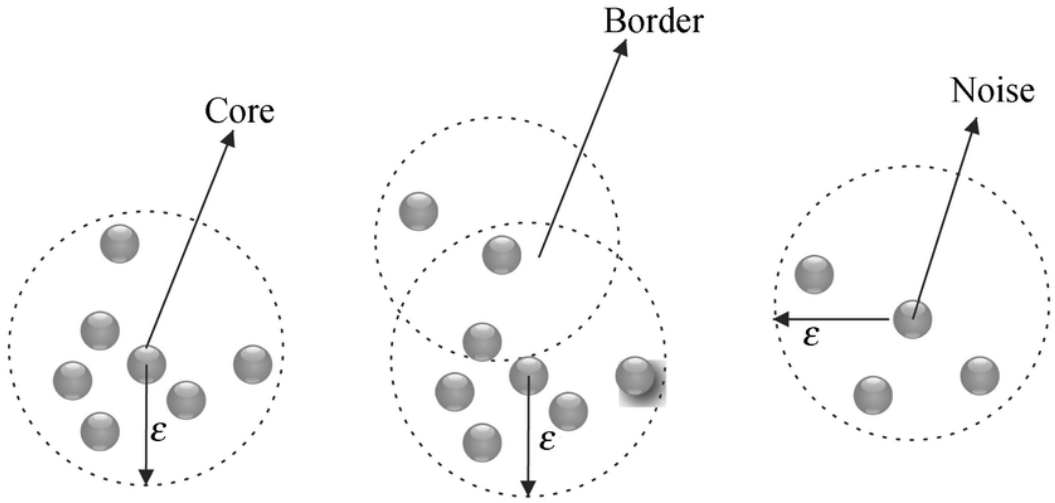
DENSITY-BASED CLUSTERING

Density-based Clustering

- Density-Based Methods
  - Clusters – Dense regions of objects separated by regions of low density
  - A cluster is defined as a maximal set of density-connected points.
  - Discovers clusters of arbitrary shape
- Approach
  - The set of points from one cluster is spatially connected
  - For any point in a cluster, the local point density around that point has to exceed some threshold
- Algorithms
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99)
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

DBSCAN (Density Based Spatial Clustering of Applications with Noise)

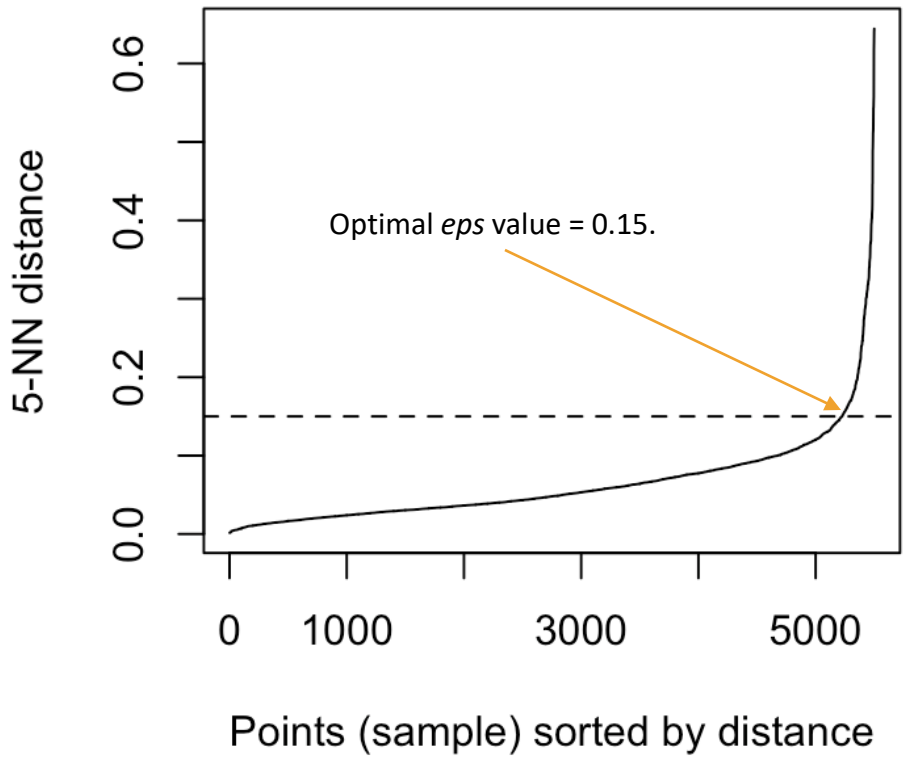
- DBSCAN computes nearest neighbor graphs and creates arbitrary-shaped clusters
- Definitions
  - Density is defined as number of points within a specified radius ε
  - A **core point** has more than a specified number of points ($MinPts$) within radius $Epsilon$ (ε)
  - A **border point** has fewer than $MinPts$ within ε, but is in the neighborhood of a core point
  - A **noise point** is any point that is not a core point or a border point



DBSCAN: core, border, and noise points

DBSCAN: Optimal Parameter Values

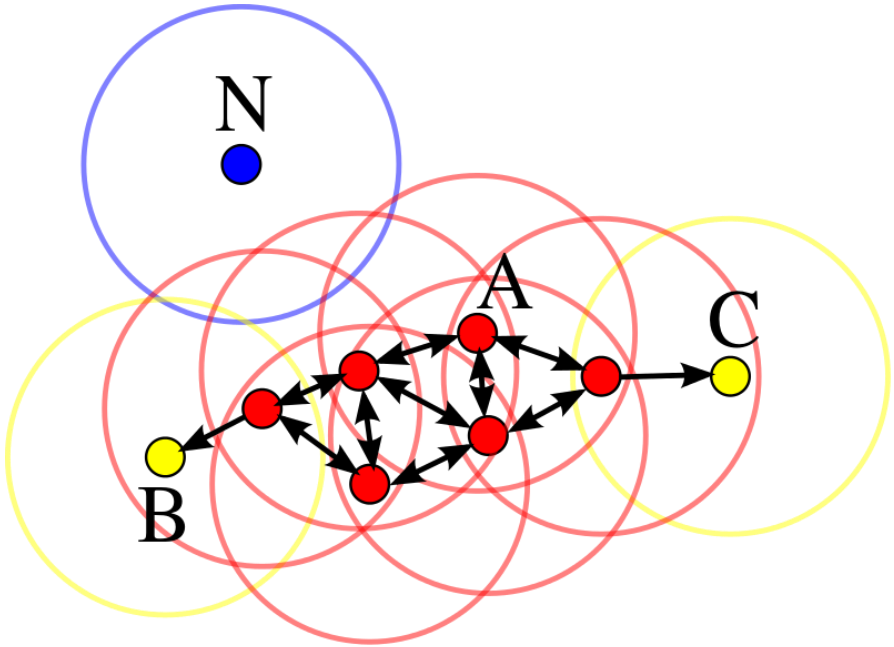
- $MinPts$
  - The larger the data set, the larger the value of $MinPts$ should be
  - Recommended: dimensions $\leq MinPts =$ 2 x dimensions
- $Epsilon\ (\varepsilon)$
  - Cannot be too small (more small clusters and noisy) or too big (large clusters)
  - Calculate the average of the distances of every point to its k nearest neighbors, where $k = MinPts$
  - Plot k-distances in an ascending order and determine the "knee", which corresponds to the optimal ε parameter.



DBSCAN: Algorithm

- Algorithm
  - Arbitrary select a point p
  - **Repeat**
    - Retrieve all density-reachable points from p with ε and $MinPts$.
    - If p is a core point, create a cluster with ε and $MinPts$.
    - If p is a border point, visit the next point
  - **Until**
    - all of the points have been visited.

  - Demo



Source: Wikipedia

DBSCAN Pros/Cons

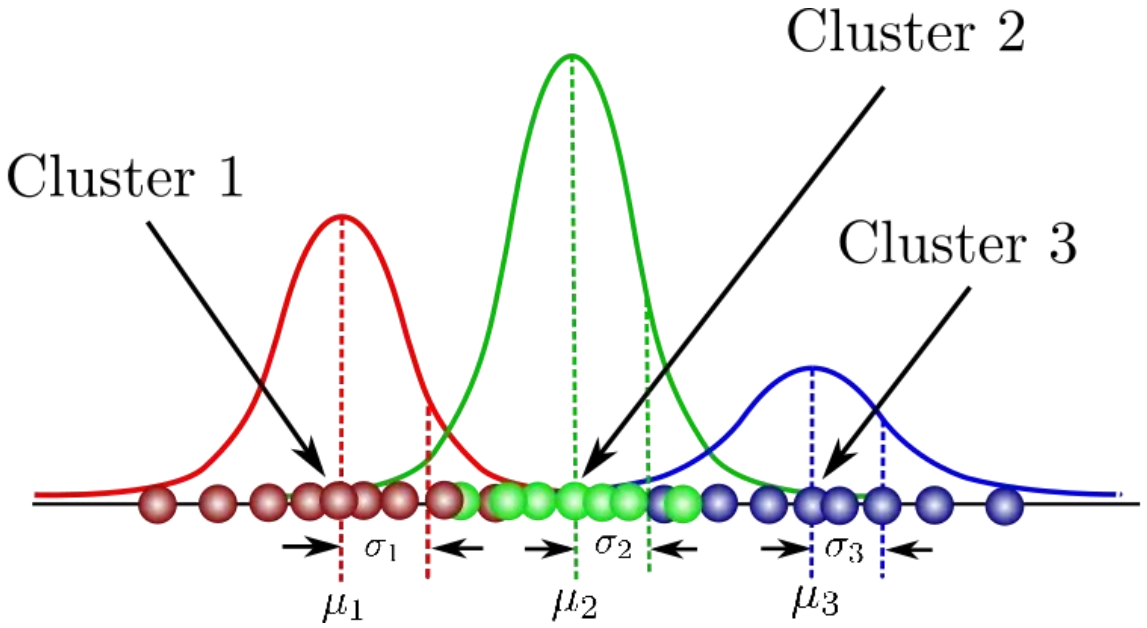
- Pros:
  - No need to specify the number of clusters
  - Flexibility in the shapes and sizes of clusters
  - Able to deal with noise and outliers
- Cons:
  - Very sensitive to input parameters ε and $MinPts$
  - Computationally expensive
  - More complicated clustering method than k-means
  - Reachable from two clusters may be arbitrarily classified
    - Not an issue for most datasets
  - Clusters of varying densities
    - Depends on a single ε value for all clusters, so clusters with variable densities may not be correctly identified
- HDBSCAN
  - Less susceptible to noise
  - More scalable

MODEL BASED CLUSTERING

Gaussian Mixture Models

Gaussian Mixture Model (GMM)

- Probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters
- Unlabeled data, unknown parameters
  - We know (assume) the number of clusters, but we do not know where these clusters are as well as how they are shaped



Gaussian Distribution

- Univariate Gaussian

$$p\left(x \mid \mu, \sigma^2\right) = \mathcal{N}\left(\mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

  - μ and σ are scalars representing the mean and standard deviation of the distribution

- Multivariate Gaussian

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = (2\pi)^{-\frac{D}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

  - μ is vector of means, and Σ is a matrix (of variances).
  - $|\Sigma|$ is the determinant of Σ, and D is the number of dimensions $x \in \mathbb{R}^D$

GMM: Expectation-Maximization (EM)

- Probabilistic model-based clustering
  - EM gives an efficient method at estimating the parameter through MLE by repeatedly constructing a lower-bound on the likelihood (E-step) and optimizing that lower-bound(M-step)
  - EM approaches maximum likelihood (MLE) or maximum a posteriori (MAP) estimates of parameters in statistical models
- EM Framework
  - Randomly initialize model parameters, then alternate between
  - **Repeat**
    - E-Step: assigns objects to clusters according to the current parameters of probabilistic clusters
    - M-Step: finds new parameters that maximize the SSE in fuzzy clustering or the expected likelihood in probabilistic model-based clustering on fully observed data.
  - **Until** convergence

GMM: Expectation-Maximization (EM)

- Step 1 (Init):
  - Initialize the parameters μ_k, π_k, σ_k to random values
- Step 2 (E-step):
  - Using current values of μ_k, π_k, σ_k to random values evaluate posterior distribution r_{nk} for each component and data point

$$r_{nk} = \frac{\pi_k \mathcal{N}\left(x_n \mid \mu_k, \sigma_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(x_n \mid \mu_j, \sigma_j\right)}$$

- Step 3 (M-step):
  - Using distributions found in step 2 evaluate parameters μ_k, π_k, σ_k
  - Univariate case

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} x_n, \quad \sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (x_n - \mu_k)^2, \quad \pi_k = \frac{N_k}{N},$$

  - Multivariate case

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n, \quad \boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \left(\boldsymbol{x}_n - \boldsymbol{\mu}_k\right) \left(\boldsymbol{x}_n - \boldsymbol{\mu}_k\right)^{\top}, \quad \pi_k = \frac{N_k}{N}.$$

- Step 4 (Validate):
  - Compare negative log-likelihood and compare it against a threshold

$$-\sum_{n=1}^{N} \left[\log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{(x_n - \mu)^2}{2\sigma^2} \right] < \epsilon$$

GMM: Cluster Shapes

- The clusters can have all shapes a multivariate Gaussian distribution can take i.e., model all ellipsoid shapes