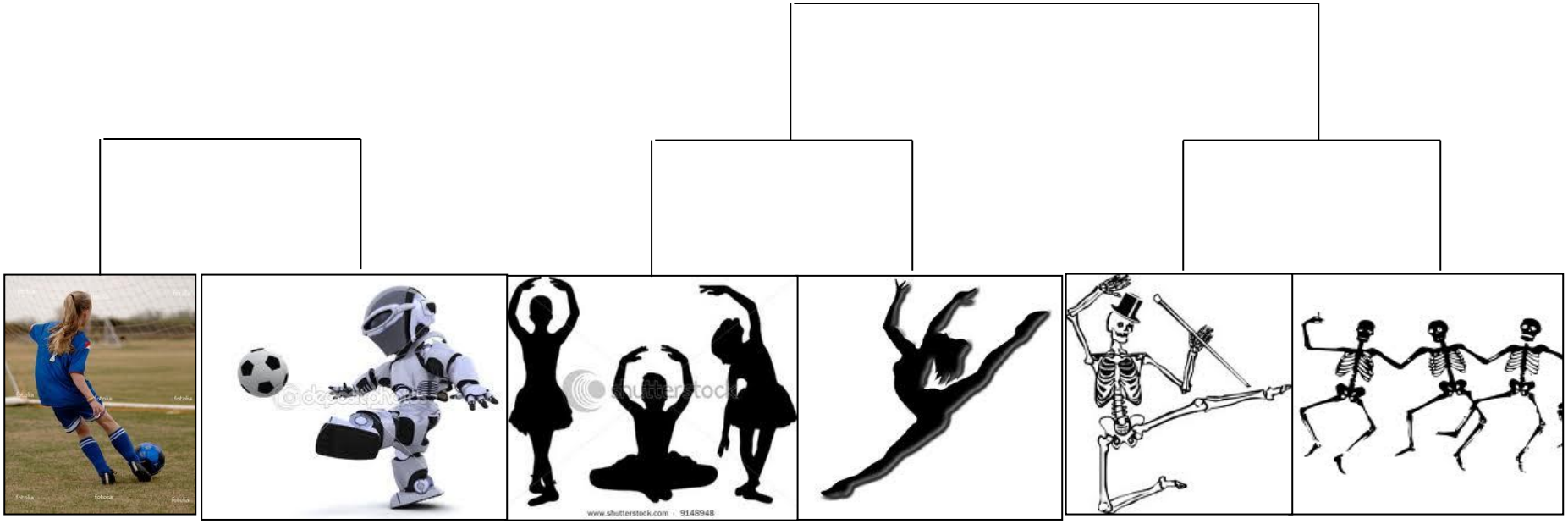
Hierarchical Clustering

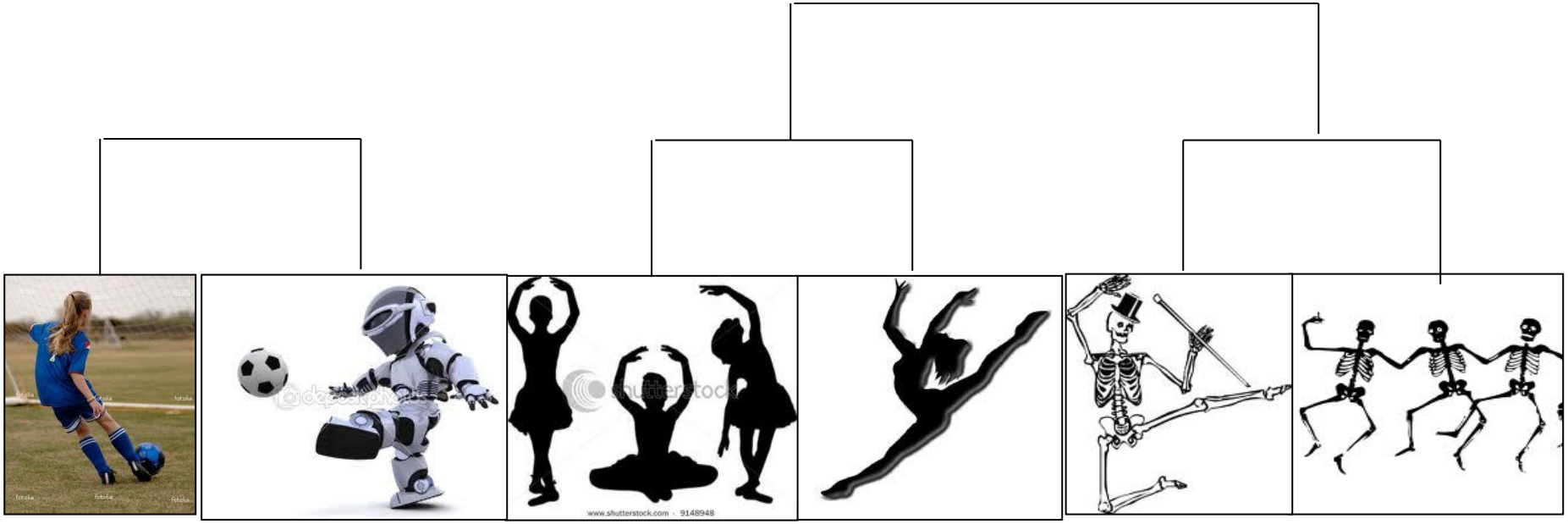
Hierarchical Clustering: Motivation



Clusters found through density-based methods (K-means, DBSCAN) are *flat.*

Data may share several features in combination or hierarchically.

Hierarchical Clustering: Principle

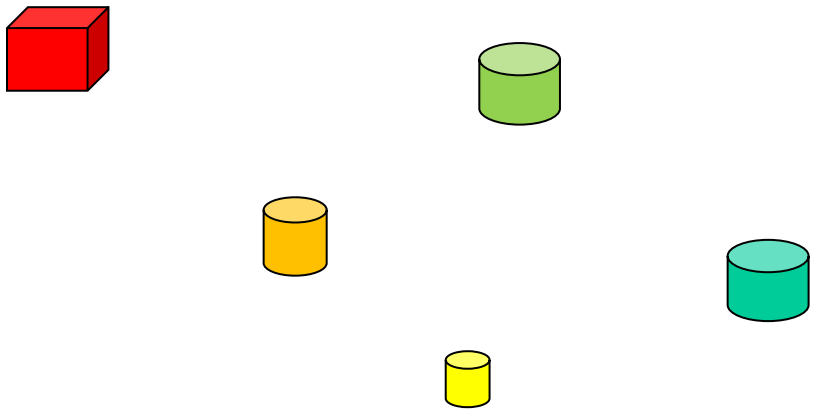


Hierarchical clustering aims at automatically determining:

- Number of groups / clusters
- Type of grouping
- Parent-child relationships across groups

Hierarchical Clustering: Algorithm

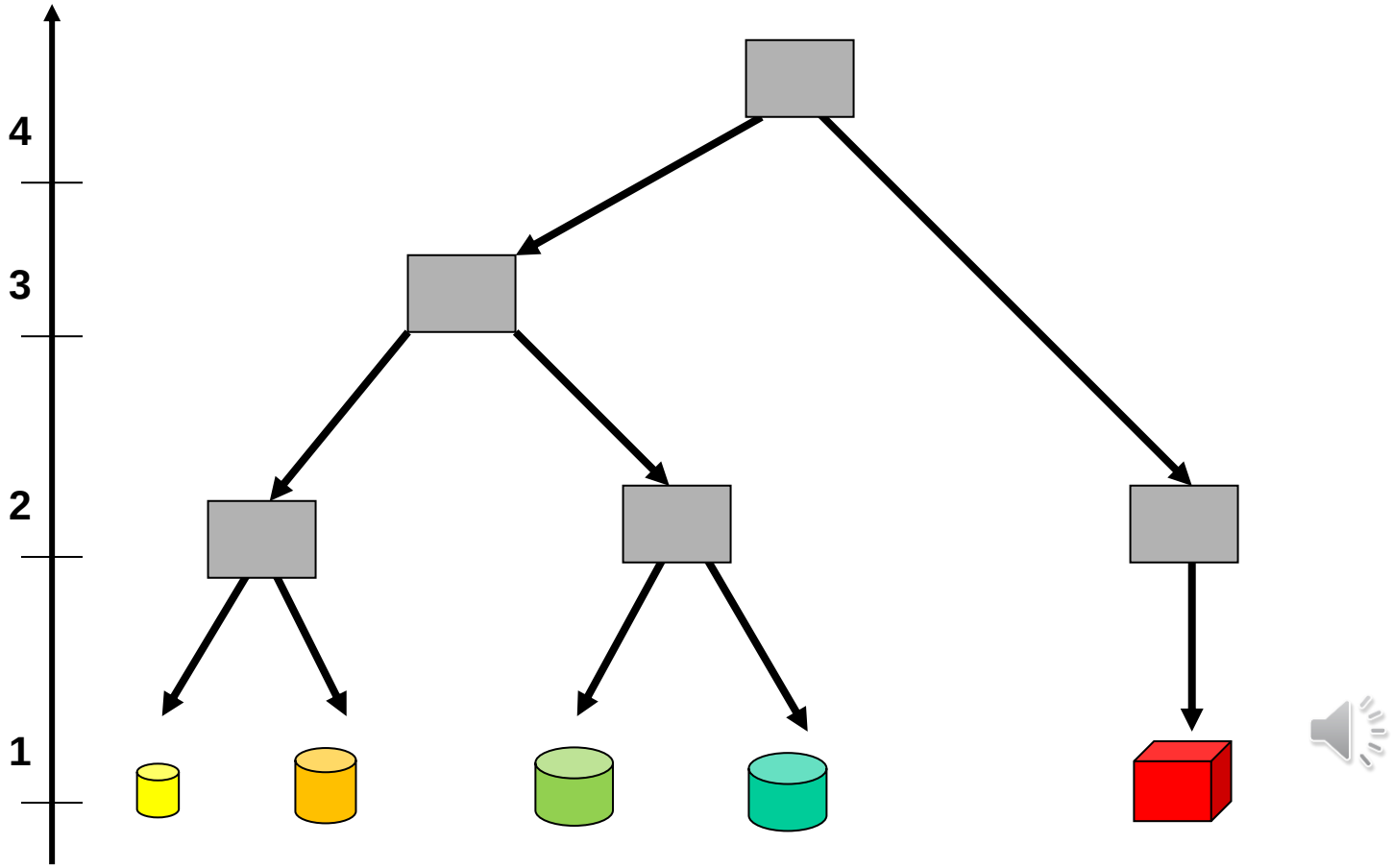
Observe these objects. They share many common features.



Hierarchical clustering groups datapoints recursively,
by either **agglomerating** or **dividing** the dataset

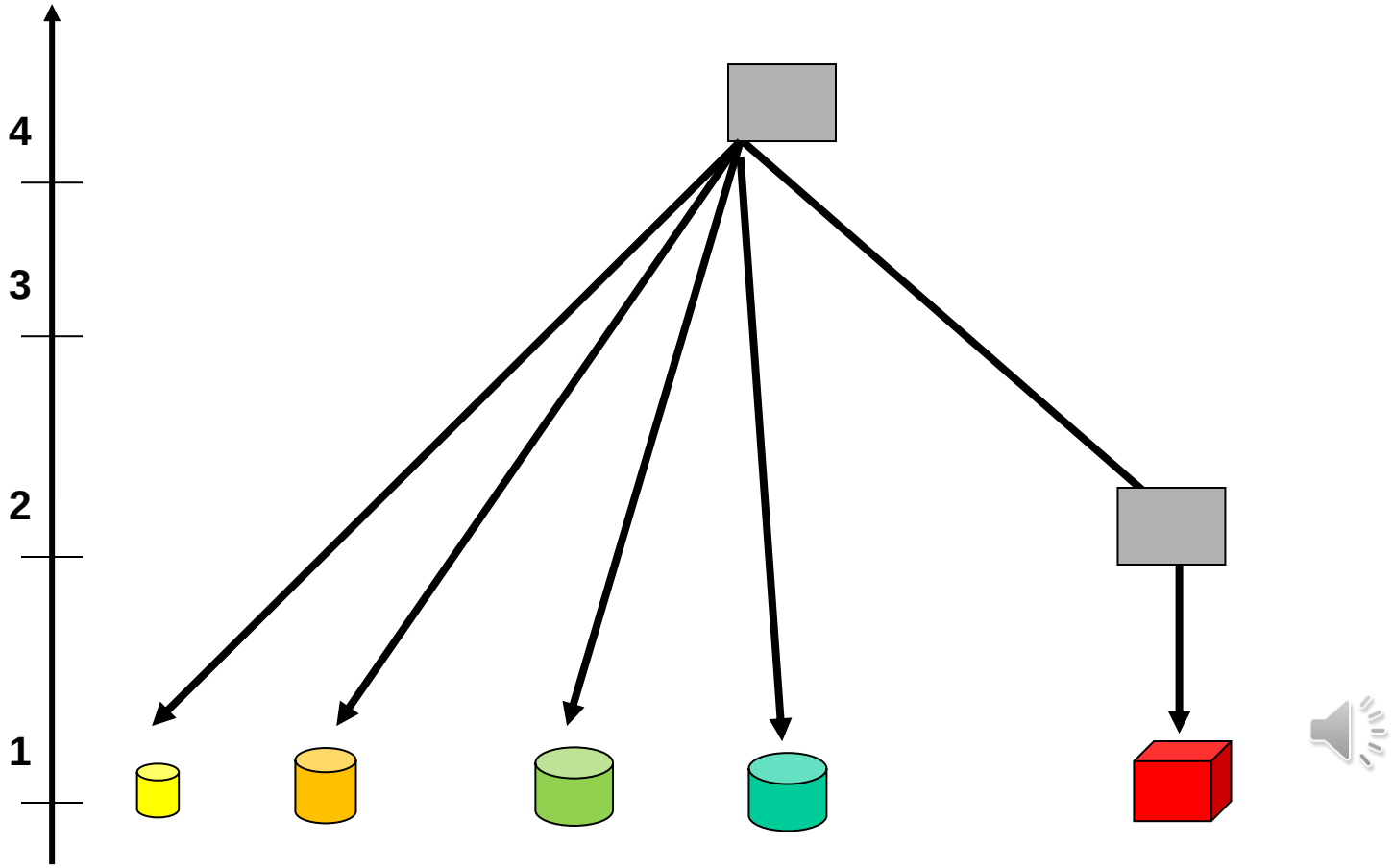
Hierarchical Clustering

In *Hierarchical Clustering*, the data is partitioned iteratively, by agglomerating the data → generates a dendogram



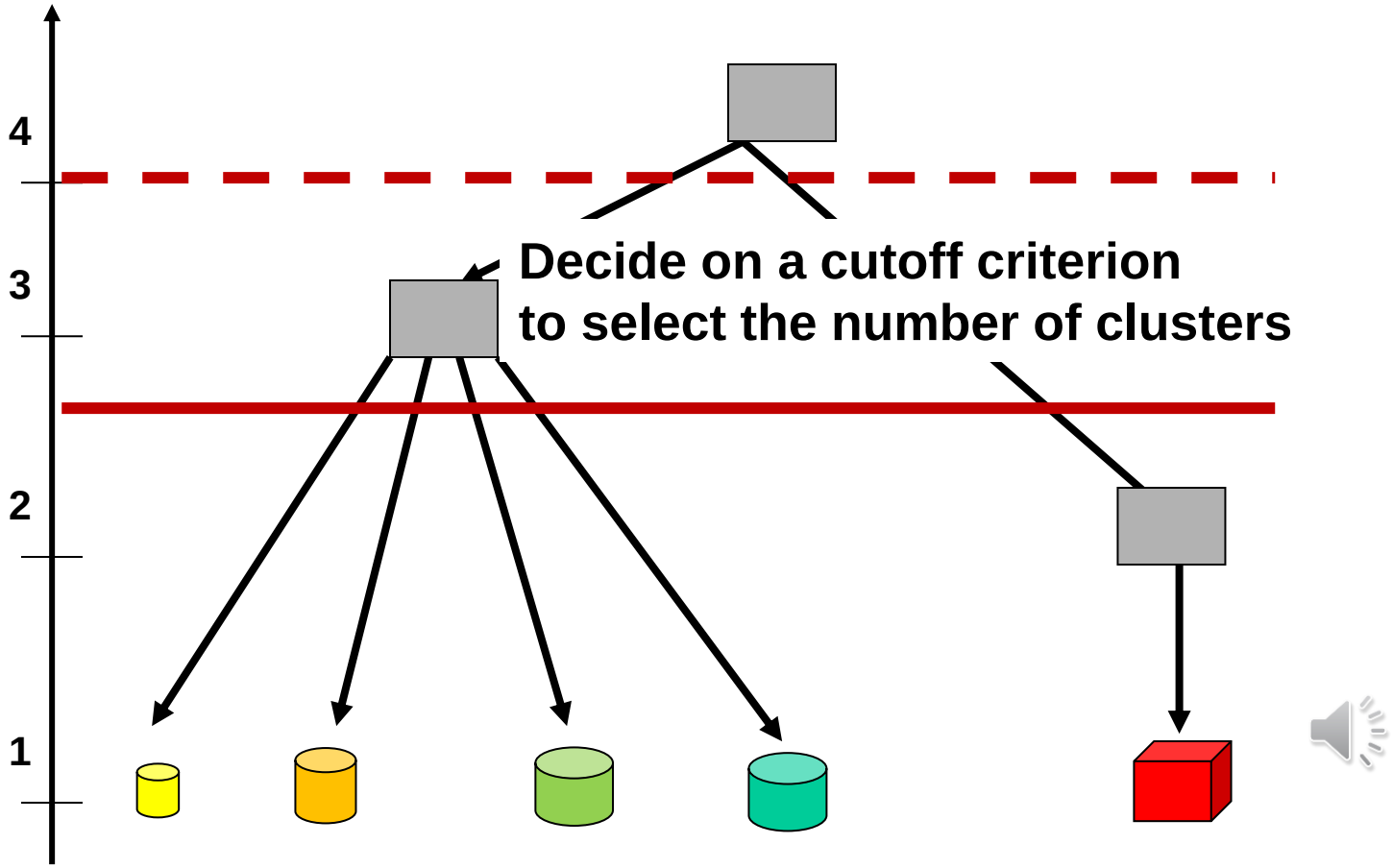
Hierarchical Clustering

The data can also be partitioned iteratively, by dividing the data from one single linkage, to multiple clusters.



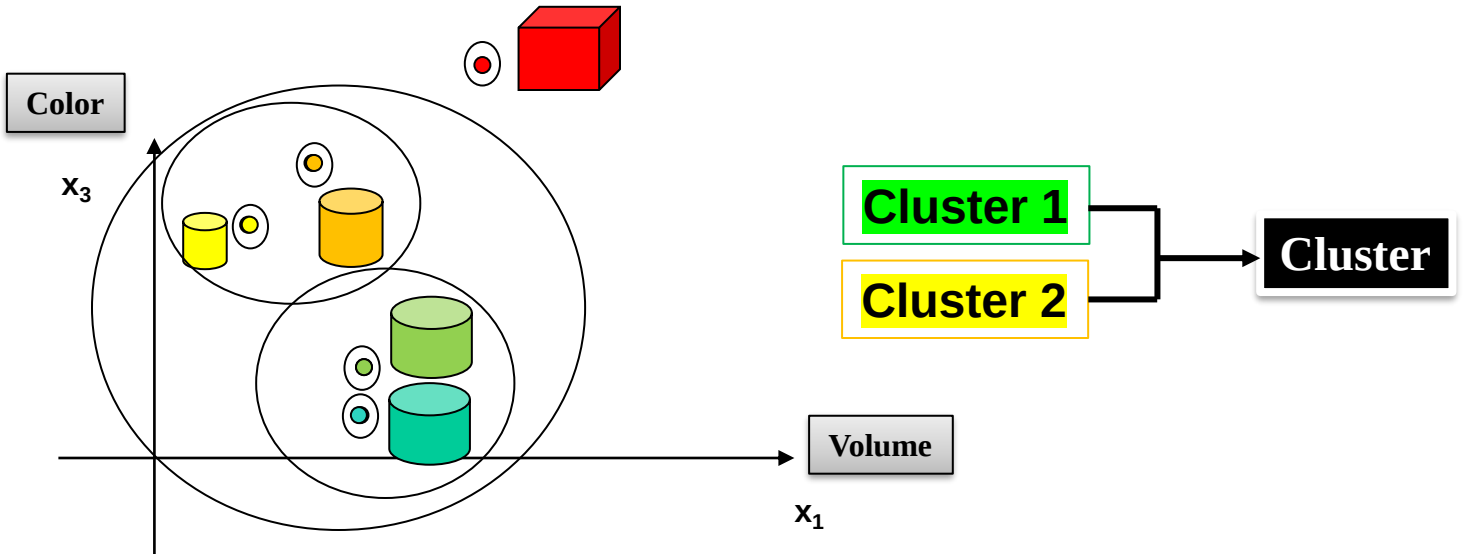
Hierarchical Clustering

The data can also be partitioned iteratively, by dividing the data from one single linkage, to multiple clusters.



Step 1: Each point is a cluster

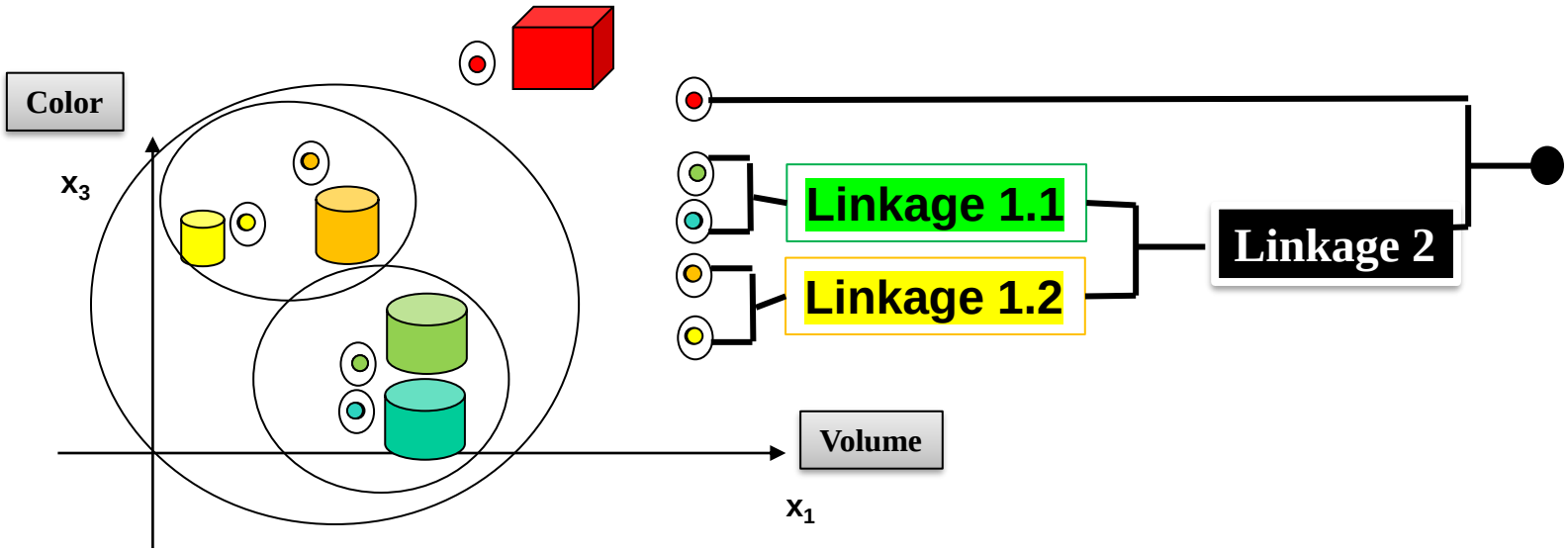
Step 2: Group points close to one another



DBSCAN groups clusters that "resemble" (are close to) each other, by merging clusters incrementally.

Step 1: Each point is a cluster

Step 2: Group points close to one another



Hierarchichal Clustering differs from DBSCAN: datapoints and clusters are not merged but linked. Linkage across datapoints and clusters is what matters!

Hierarchical Clustering: Similarity Metrics

The similarity metric $d(x, y)$ across two datapoints x and y has the following properties:

(1) symmetric $d(x, y) = d(y, x)$

(2) positivity $d(x, y) \geq 0$ and $d(x, y) = 0$ iff $x = y$

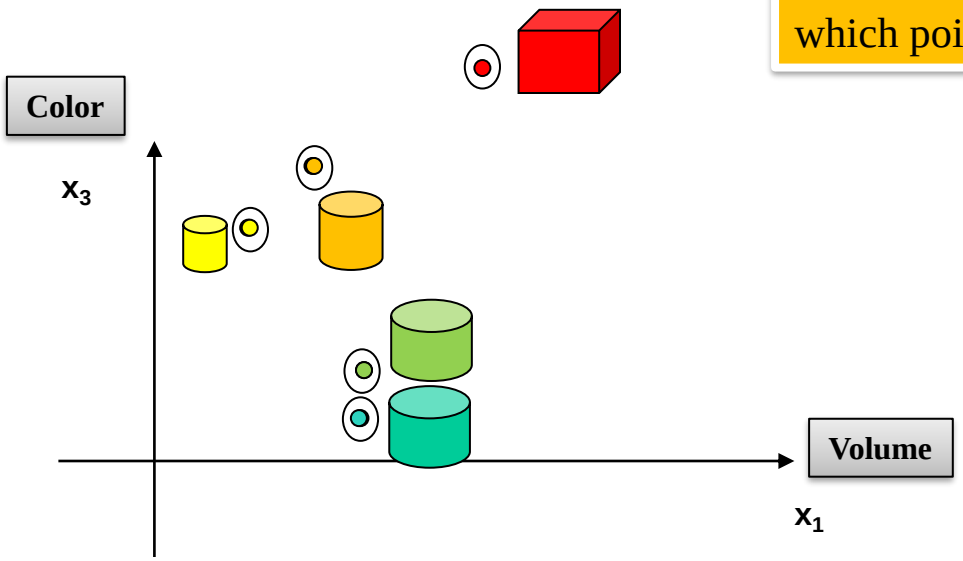
(3) triangle inequality $d(x, z) \geq d(x, y) + d(y, z), \ \forall x, y, z.$

Distance measure	Calculation formula
Euclidean:	$d(x, y) = \sqrt{\sum_{i=1}^{N} \lvert x_i - y_i \rvert^2}, \ i = 1..., N \ \& \ x, \ y \in \mathbb{R}^N$
Manhattan:	$d(x, y) = \sum_{i=1}^{N} \lvert x_i - y_i \rvert$
Maximum distance:	$d(x, y) = \max\left(\lvert x_i - y_i \rvert\right)$
Cosine distance:	$d(x, y) = \frac{\sum_i x_i y_i}{\lVert x \rVert \lVert y \rVert}$
Mahalanobis distance:	$d(x, y) = (x - y)^T S^{-1} (x - y)$ S: within-sample covariance matrix

Step 1: Each point is a cluster

Step 2: Group points close to one another

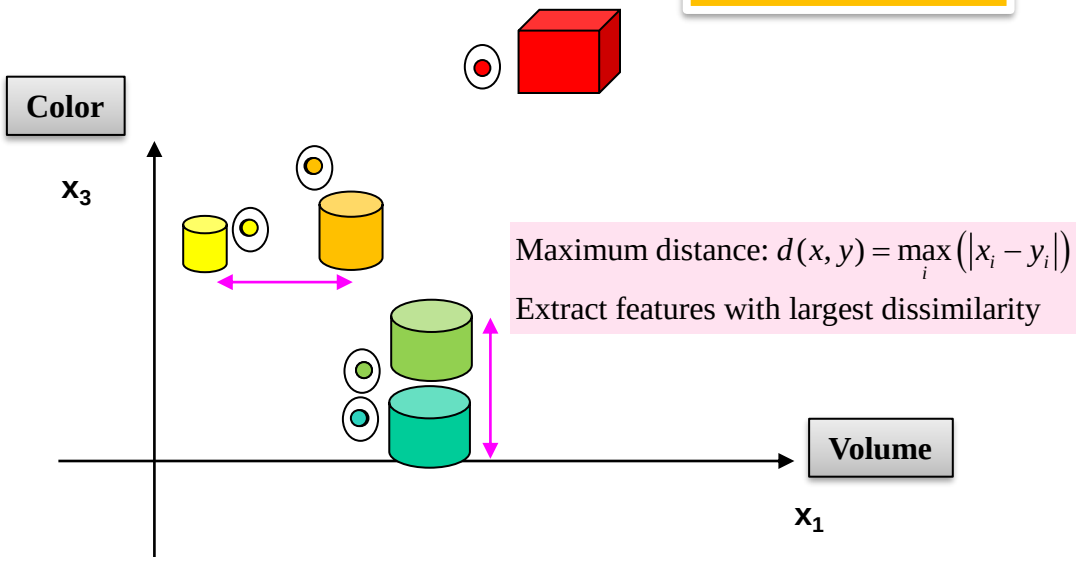
Need a metric to determine
which points are **close** to one another



Step 1: Each point is a cluster

Step 2: Group points close to one another

Popular metrics



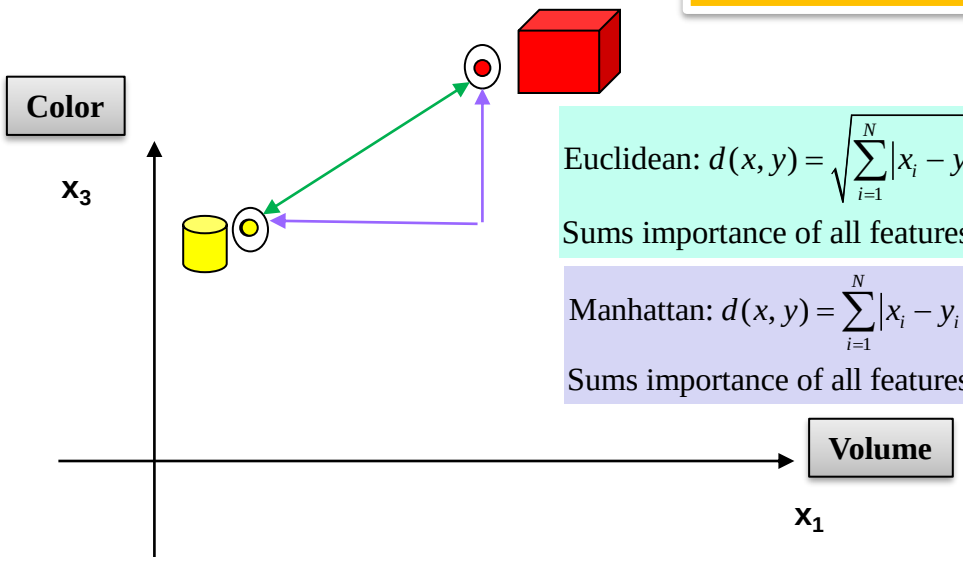
Maximum distance: $d(x, y) = \max_i \left(|x_i - y_i| \right)$

Extract features with largest dissimilarity

Step 1: Each point is a cluster

Step 2: Group points close to one another

Popular metrics



Euclidean: $d(x, y) = \sqrt{\sum_{i=1}^{N} |x_i - y_i|^2}$, $i = 1..., N$ & $x,\ y \in \mathbb{R}^N$

Sums importance of all features, sensitive to absolute value of each scale.

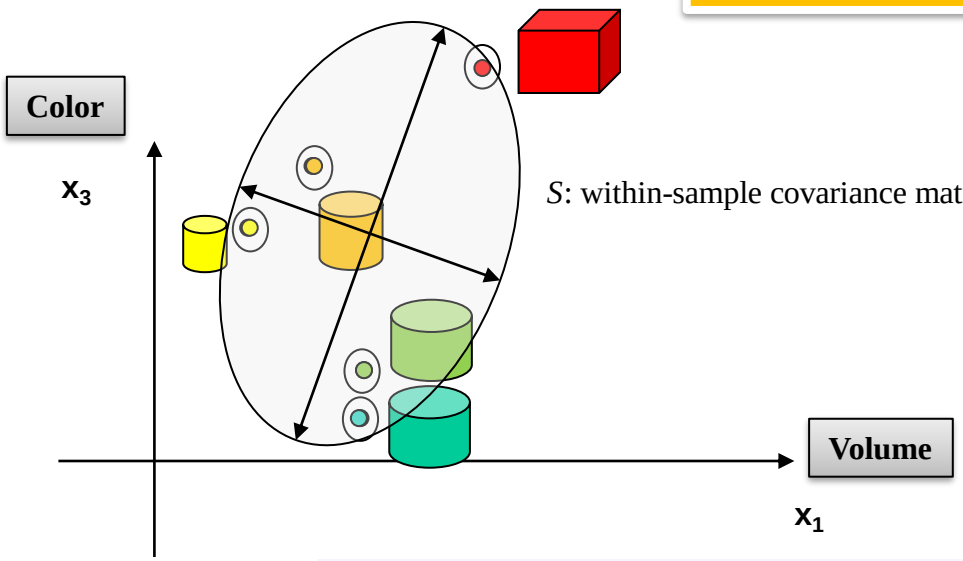
Manhattan: $d(x, y) = \sum_{i=1}^{N} |x_i - y_i|$

Sums importance of all features, sensitive to absolute value of each scale.

Step 1: Each point is a cluster

Step 2: Group points close to one another

Popular metrics



S: within-sample covariance matrix

Mahalanobis distance: $d(x,y) = (x-y)^T S^{-1} (x-y)$

Normalizes for relative spread of each feature

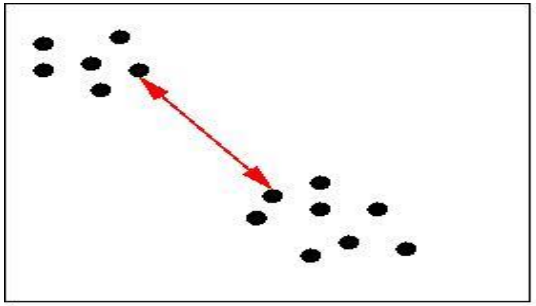
Sums relative contribution of each feature with respect to variance of whole dataset;

Insensitive to the scale of each feature.

Agglomerative method - Algorithm

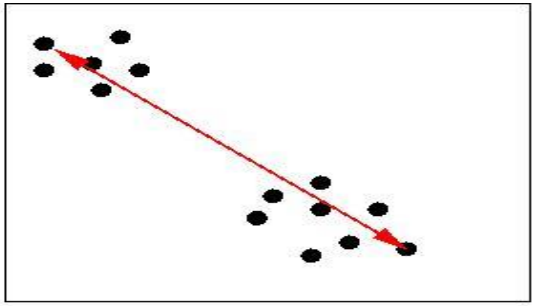
1. **Initialization:** To each of the *M* data points $x^i, i = 1...M$ associate one cluster c_i. You, thus, start with *M* clusters.

2. Find the closest clusters according to a *distance metric* $d\left(c_i, c_j\right)$ The distance between groups can either be:



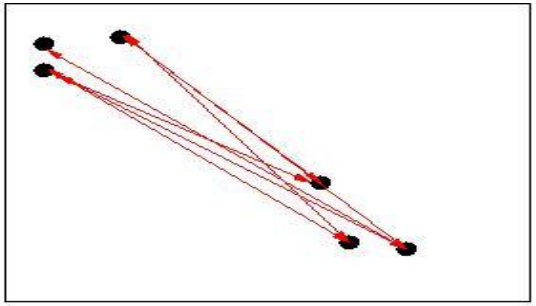
Single Linkage

$$d\left(c_i, c_j\right) = \min_{x^k \in c_i, x^l \in c_j} d\left(x^k, x^l\right)$$



Complete Linkage

$$d\left(c_i, c_j\right) = \max_{x^k \in c_i, x^l \in c_j} d\left(x^k, x^l\right)$$



Average Linkage

$$d\left(c_i, c_j\right) = \operatorname{mean}_{x^k \in c_i, x^l \in c_j} d\left(x^k, x^l\right)$$

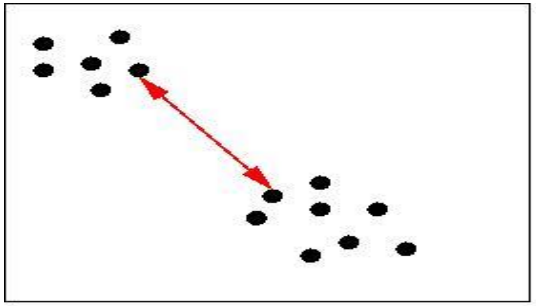
3. Create a new cluster to encompass all datapoints in the previous cluster.

4. Stops once all data are linked through a single cluster.

Divisive method - Algorithm

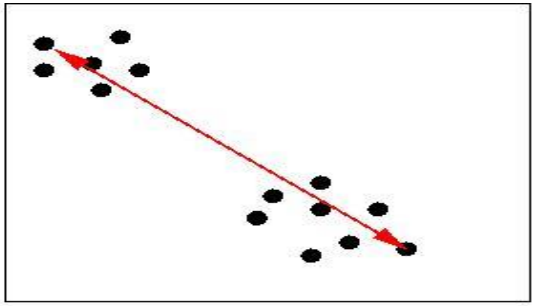
1. **Initialization:** To all the *M* data points $x^i, i = 1...M$ associate one single cluster c_1 .

2. Find the points farthest apart according to a *distance metric* $d(c_i, c_j)$ The distance between groups can either be:



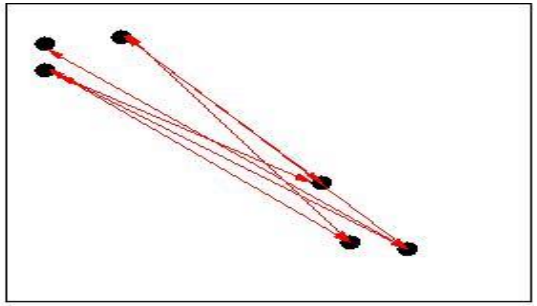
Single Linkage

$$d(c_i, c_j) = \min_{x^k \in c_i, x^l \in c_j} d(x^k, x^l)$$



Complete Linkage

$$d(c_i, c_j) = \max_{x^k \in c_i, x^l \in c_j} d(x^k, x^l)$$



Average Linkage

$$d(c_i, c_j) = \text{mean}_{x^k \in c_i, x^l \in c_j} d(x^k, x^l)$$

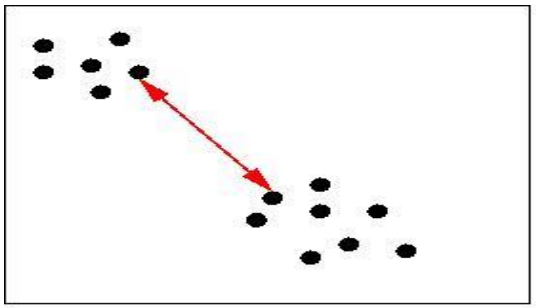
3. Divide the points into two new clusters according to a cutoff measure on the distance between points.

4. Stops once each datapoint is associated a single cluster.

Divisive method - Algorithm

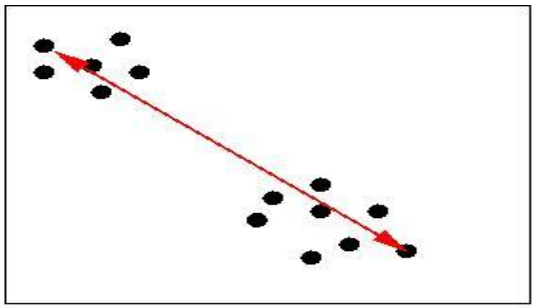
1. **Initialization:** To all the *M* data points $x^i, i = 1...M$ associate one single cluster c_1 .

2. Find the points farthest apart according to a *distance metric* $d(c_i, c_j)$ The distance between groups can either be:



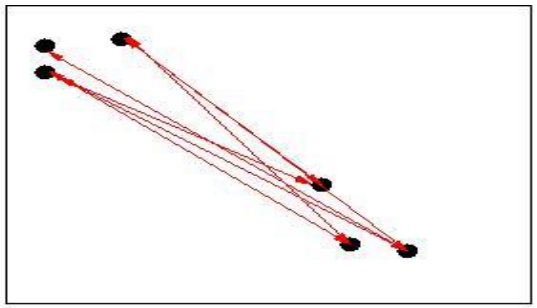
Single Linkage

$$d(c_i, c_j) = \min_{x^k \in c_i, x^l \in c_j} d(x^k, x^l)$$



Complete Linkage

$$d(c_i, c_j) = \max_{x^k \in c_i, x^l \in c_j} d(x^k, x^l)$$



Average Linkage

$$d(c_i, c_j) = \text{mean}_{x^k \in c_i, x^l \in c_j} d(x^k, x^l)$$

"Chaining"
Sequence of close observations in different groups cause early merges of those groups

Might not merge close groups because outlier members are too far apart

Best tradeoff, but more intensive computationally and depends on previous grouping.

Hierarchical Clustering: Example

Hierarchical clustering can be used with arbitrary sets of data.

Example:

Hierarchical clustering to discover similar temporal pattern of crimes across districts in India.

Chandra et al, "A Multivariate Time Series Clustering Approach for Crime Trends Prediction", IEEE SMC 2008.

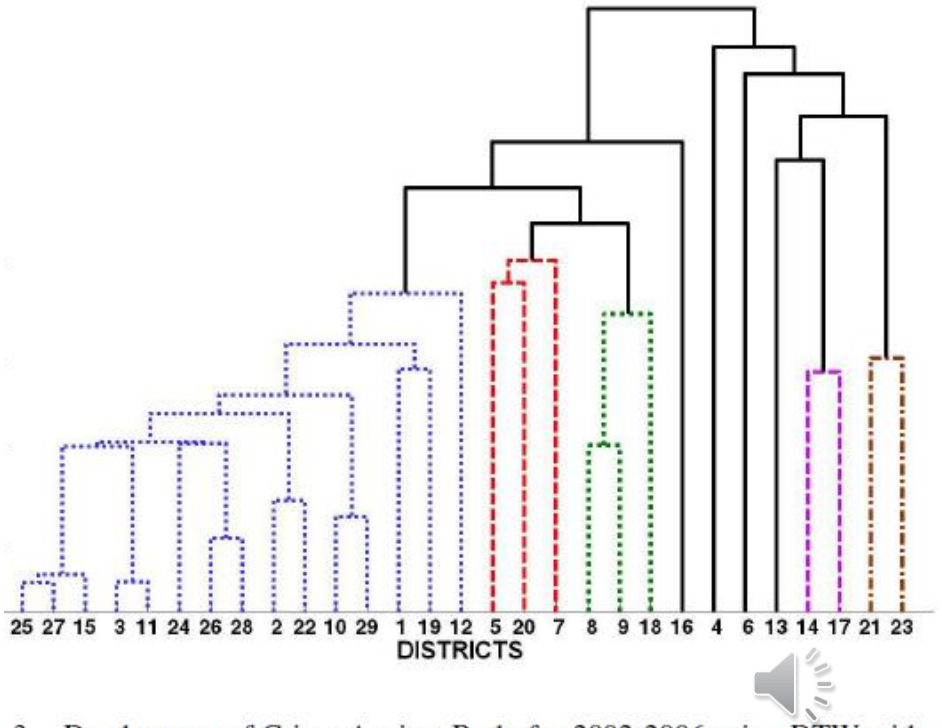


Fig. 3. Dendrogram of Crime Against Body for 2002-2006 using DTW with Parametric Minkowski Model

Hierarchical Clustering: Limitations

- Once a merging or division is done on one level of the hierarchy, it cannot be undone.

- It is costly both in computation time and memory, especially for large scale problems. Generally, the time complexity of hierarchical clustering is quadratic $O(M^2)$ about the number of data points (M) clustered.

Hierarchical Clustering: Recent Advances

New Distance Functions:

- New techniques study the graph-based structure of hierarchical clustering and use graph theory to decide on similarity across two graphs or subgraphs.

- Density-based approaches such as GMM can be used to estimate the distribution of each cluster, as a measure of similarity.

New Update Rules:

- New approaches consider how to dynamically update the structure of the tree or of sub-branches of the tree as new data arise.

- To decrease computational costs, various options are looked at to represent the tree, e.g. a coarser representation of clusters through centroids, akin to what is used in K-means, lead to linear growth in place of quadratic growths