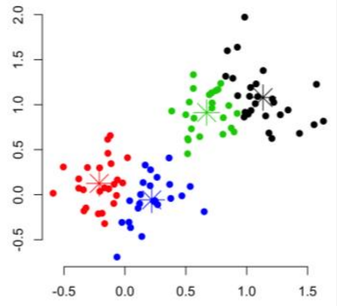
3141 - Machine Learning from Data

Spring 2022



Ariel Shamir

Zohar Yakhini

Leon Anavy

Unsupervised Learning and Clustering Algorithms

Unsupervised Learning

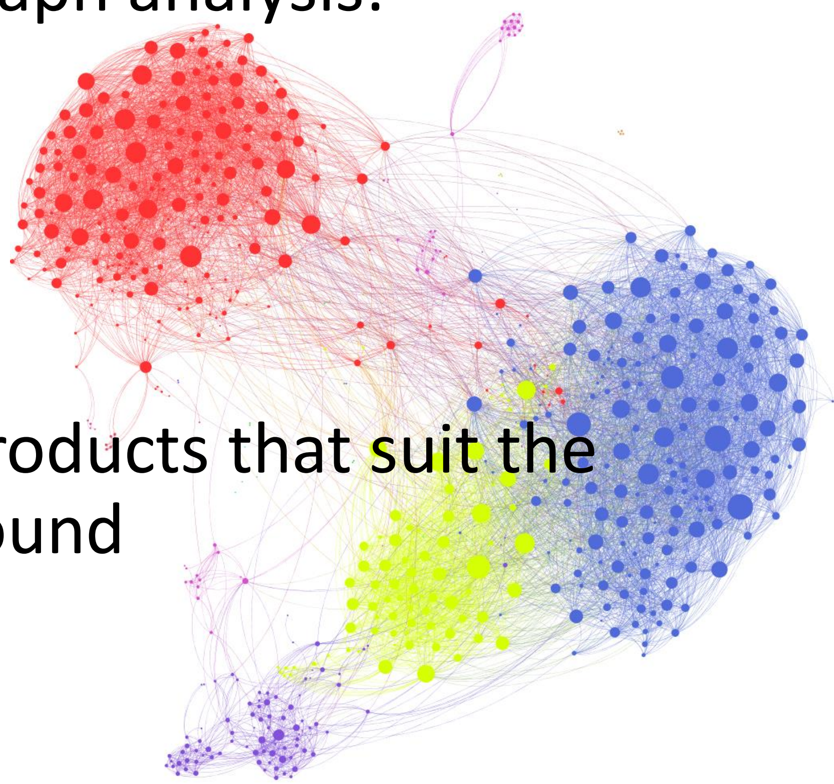
- So far, we used the labeled training data (value, class)
- The labeled served as guides to the learning process
- Using unlabeled data results in “unsupervised” learning
- What can be learned with no “target function”?
  - Structure
  - Regularity
  - Similarities
  - Grouping
- Clustering is a common unsupervised learning task

Clustering

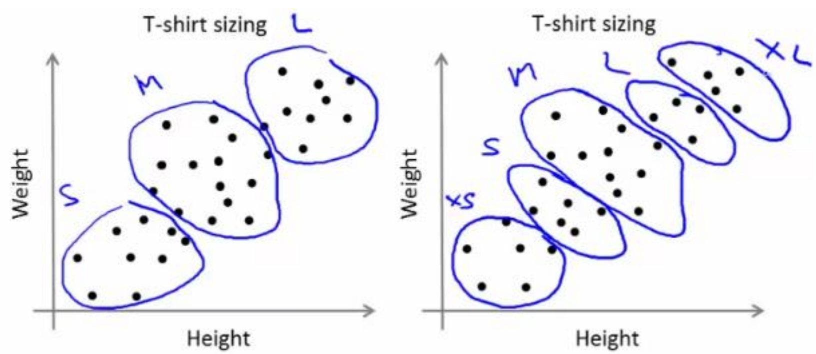
- Use the features in the data to segment/separate the samples into clusters (groups)
- Elements in each group are supposed to be “more similar” to each other than to elements in other groups
- **Similarity** and how you measure it is key in clustering!
- The method of how to create the clusters (clustering algorithm) is also a key factor

Examples for Using Clustering

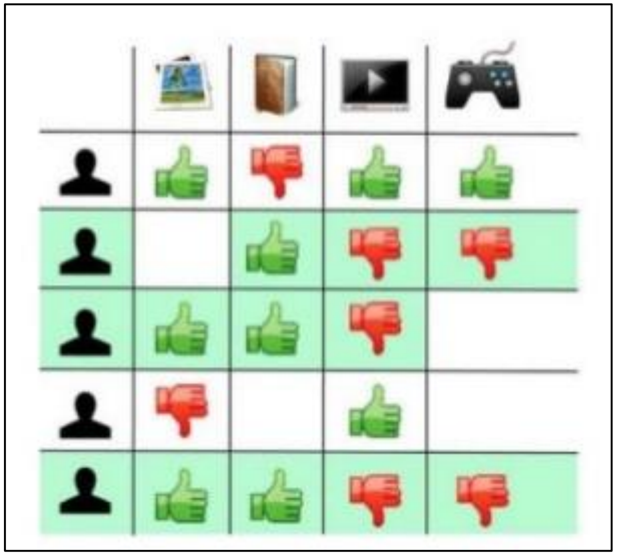
- Find clusters in network and graph analysis:
  - Social networks
  - Protein interaction
  - Similarity of audio tracks/users



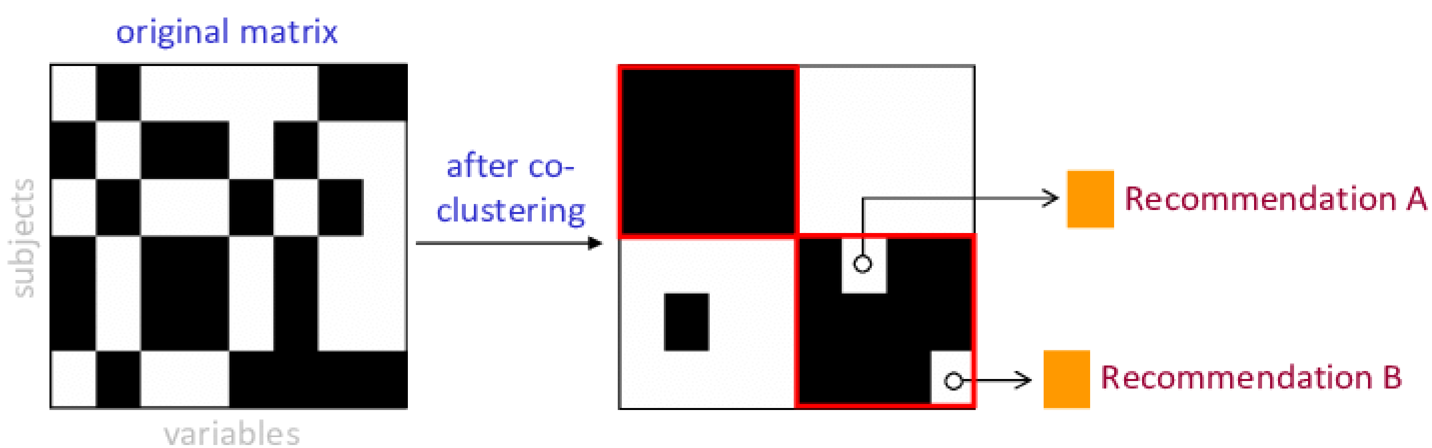
- Market segmentation - build products that suit the needs of the subpopulations found



Clustering Applications: Recommendation Systems



- Cluster items & users
- Recommend according to partners in the same cluster



Similarity Measures

- Clustering can be viewed as finding the ‘natural’ groupings in a data set.
- How can we say that samples in one cluster are more similar to each other than they are to samples in another cluster?

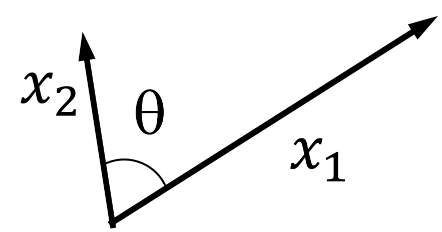
- This involves two major issues:
  - What is the **measure** of similarity between samples?
  - How can we **evaluate** the partitioning of the set into clusters?

Similarity Measures

- A symmetric function whose value is large when two instance vectors are more similar
- We can use a distance measure to derive similarity: the closer the instances are, the more similar they are

- Examples:
  - Euclidean
  - Minkowski metric
  - Normalized inner product
  - Pi minus the angle between vectors
  - Others…

$$S(x_1, x_2) = Cos(\theta) = \frac{x_1^T x_2}{\|x_1\| \|x_2\|}$$



Distance Metric (Function)

- Similarity is often defined as the inverse of the distance
- A distance metric on a set X is a non-negative function $d: X \times X \to [0, \infty)$ (= Non-negativity: $d(x_1, x_2) \geq 0$)

- Satisfying for all $x_1, x_2, x_3 \in X$ the following:
  - Identity of indiscernibles
  $$d(x_1, x_2) = 0 \iff x_1 = x_2$$
  - Symmetry:
  $$d(x_1, x_2) = d(x_2, x_1)$$
  - Triangle inequality:
  $$d(x_1, x_2) \leq d(x_1, x_3) + d(x_3, x_2)$$

Popular Distance Functions

Herman Minkowski
1864-1909

For $x, y \in \mathbb{R}^n$

$x = (x_1, x_2, \dots, x_n), y = (y_1, y_2, \dots, y_n)$

- Manhattan distance:

$$d_{L_1}(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

- Euclidean distance:

$$d_{L_2}(x, y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} = \|x - y\|$$

- L_p Minkowski distance:

$$d_{L_p}(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}, \qquad p \geq 1$$

- Infinity norm:

$$d_{L_\infty}(x, y) = \max_i(|x_i - y_i|)$$

Naïve Algorithm: Cluster Growing

- Simple straight-forward algorithm

while there are still un-clustered elements in data
do

  pick a seed element s and create the cluster $\mathbf{C_S}$

  mark s as clustered

  while there is an un-clustered element e with $\mathbf{d(e, C_s)} < T$
  do

    insert e to $\mathbf{C_s}$

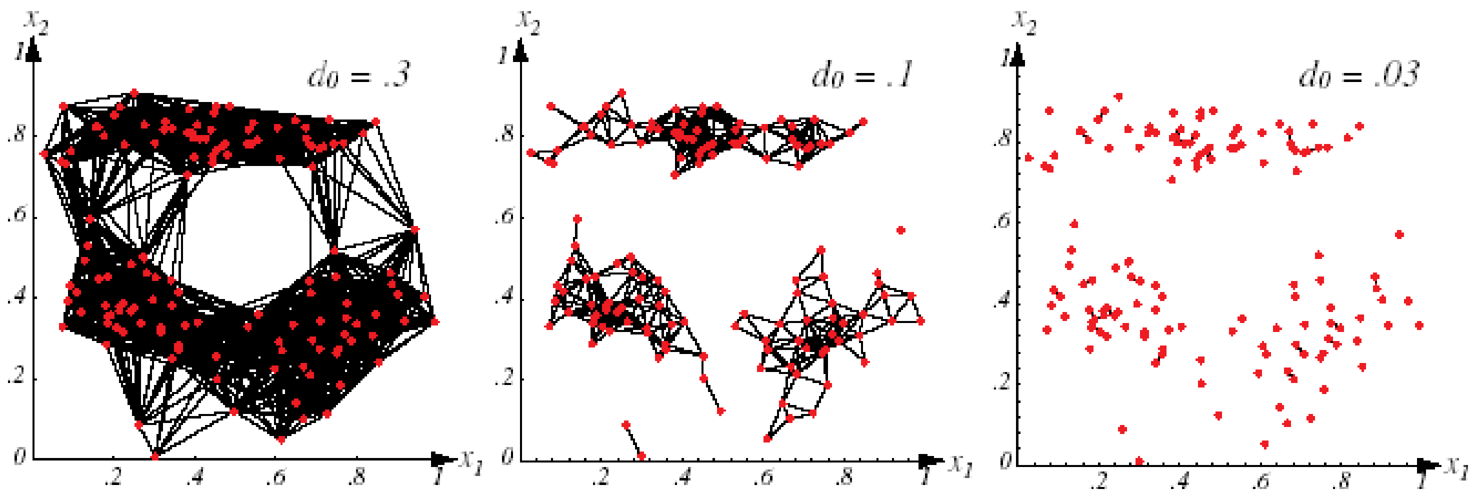
    mark e as clustered

  end

end

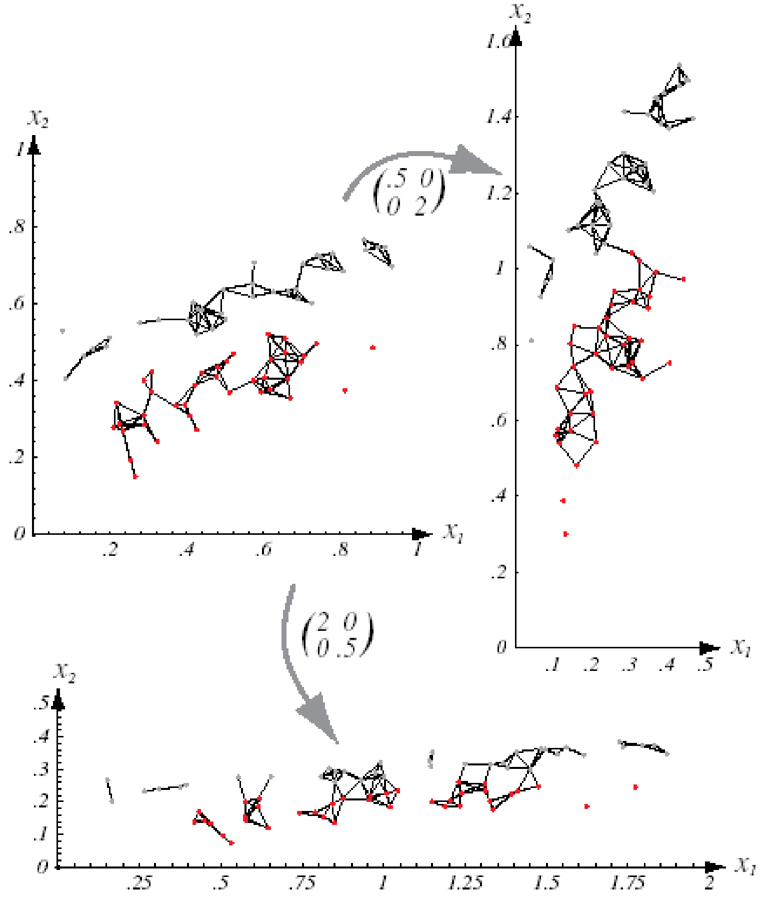
Hyperparameter: the threshold T

Different Ts may lead to different results



Clustering Invariance

- The clusters depend on the choice of a particular similarity measure and its properties
- Euclidean distance is invariant to rigid body transformation, but not to non-uniform scale!



Normalizing

- Using PCA and normalizing the data (unit variance) can help "invariance"
- But this is not necessarily desirable for clustering:

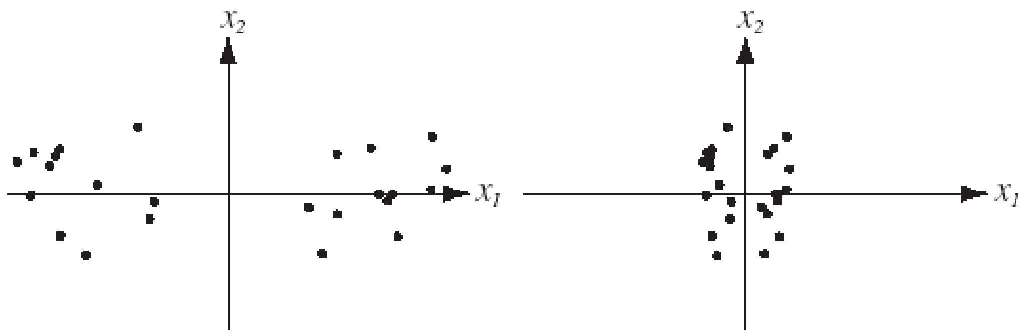


FIGURE 10.9. If the data fall into well-separated clusters (left), normalization by scaling for unit variance for the full data may reduce the separation, and hence be undesirable (right). Such a normalization may in fact be appropriate if the full data set arises from a single fundamental process (with noise), but inappropriate if there are several different processes, as shown here. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Cluster Growing Discussion

- Search/data structures: computing the graph of adjacencies takes $O(n^2)$, so we need more efficient methods to add all elements with distance < T.
- We have already seen such search structures/heuristics for KNN algorithm.
- Depends on the order of selection.
- The threshold T needs to be determined.

- How do we evaluate the result?

Clustering as an Optimization Task

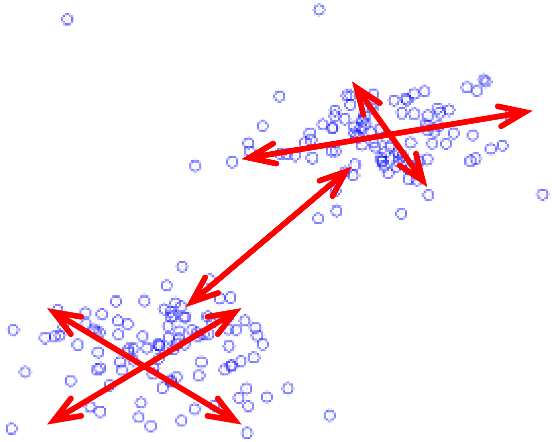
- Clustering task:
  We have a data set of samples $D = \{x_1, \ldots, x_\mathrm{m}\}$.
  Partition them into k disjoint subsets $\mathrm{C}_1, \ldots, \mathrm{C}_\mathrm{k}$.
- Challenge:
  Formulate and perform the clustering task as an **optimization** of some criterion function.
- The criterion function should take the data set D and the defined clusters as input and produce a real number

$$f(D, \{C_i\}_{i=1}^{k}) \rightarrow R$$

Two Aspects of Clustering Quality

$$f(D, \{C_i\}_{i=1}^{k})$$

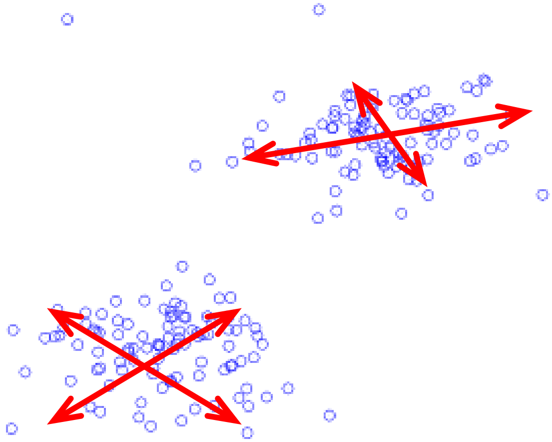
- We want to **increase the compactness** of the cloud of points **within** each one of the clusters.
- We also want to **increase the distance between** different clusters.



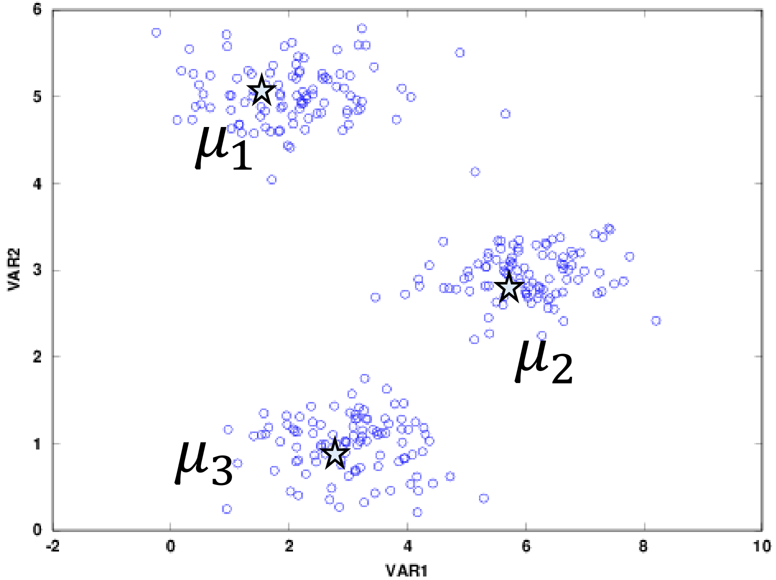
Ariel Shamir, Zohar Yakhini, Leon Anavy, IDC

First Aspects of Clustering Quality

- We want to **increase the compactness** of the cloud of points **within** each one of the clusters.
- How to measure compactness?



Centroid Based Clustering



- Each cluster is represented by its centroid:

$$\mu_i = \frac{1}{n_i}\sum_{x\in C_i} x\,, \qquad n_i = |C_i|$$

- Our objective function, under Euclidean distance, is:

$$S_C\left(D,\{C_i\}_{i=1}^{k}\right) = \frac{1}{m}\sum_{i=1}^{k}\sum_{x\in C_i}\|x-\mu_i\|^2$$

- Our goal – minimize S_C:

$$\min_{\{C_i\}_{i=1}^{k}} S_C\left(D,\{C_i\}_{i=1}^{k}\right)$$

Which evaluation criteria is addressed by this objective function?

- **Increasing the compactness** of the cloud of points **within** each one of the clusters.

$$S_C\left(D,\{C_i\}_{i=1}^{k}\right)=\frac{1}{m}\sum_{i=1}^{k}\sum_{x\in C_i}\|x-\mu_i\|^2$$

$$\min_{\{C_i\}_{i=1}^{k}} S_C\left(D,\{C_i\}_{i=1}^{k}\right)$$

- Note: we also want to **increase the distance between** different clusters. (later)

$$S_C\left(D, \{C_i\}_{i=1}^k\right) = \frac{1}{m}\sum_{i=1}^{k}\sum_{x\in C_i}\|x-\mu_i\|^2$$

Finding a Solution

$$\min_{\{C_i\}_{i=1}^k} S_C\left(D, \{C_i\}_{i=1}^k\right)$$

- Once we have a criterion function clustering becomes a well-defined computational task
- Is there an optimal solution?
- Exhaustive search one can find an optimal solution
- $\sim\frac{km}{k!}$ partitions of m elements into k clusters (the precise solutions are called Stirling numbers)
- More if we are also searching for the best k
- Possible practical approach: region/cluster growing or other heuristic search approaches.

k-Means Clustering Algorithm

- A very popular and useful clustering algorithm
- Assumes that we determined the desired number of clusters = k (this is a model hyper-parameter)
- Seeks a partitioning of the data into k (disjoint) sets, whose union is the whole data, which minimizes the Euclidean norm criterion $S_C\left(D, \{C_i\}_{i=1}^{k}\right)$.

$$S_C\left(D, \{C_i\}_{i=1}^{k}\right) = \frac{1}{m}\sum_{i=1}^{k}\sum_{x \in C_i} \|x - \mu_i\|^2$$

$$\min_{\{C_i\}_{i=1}^{k}} S_C\left(D, \{C_i\}_{i=1}^{k}\right)$$

k-Means Clustering Algorithm

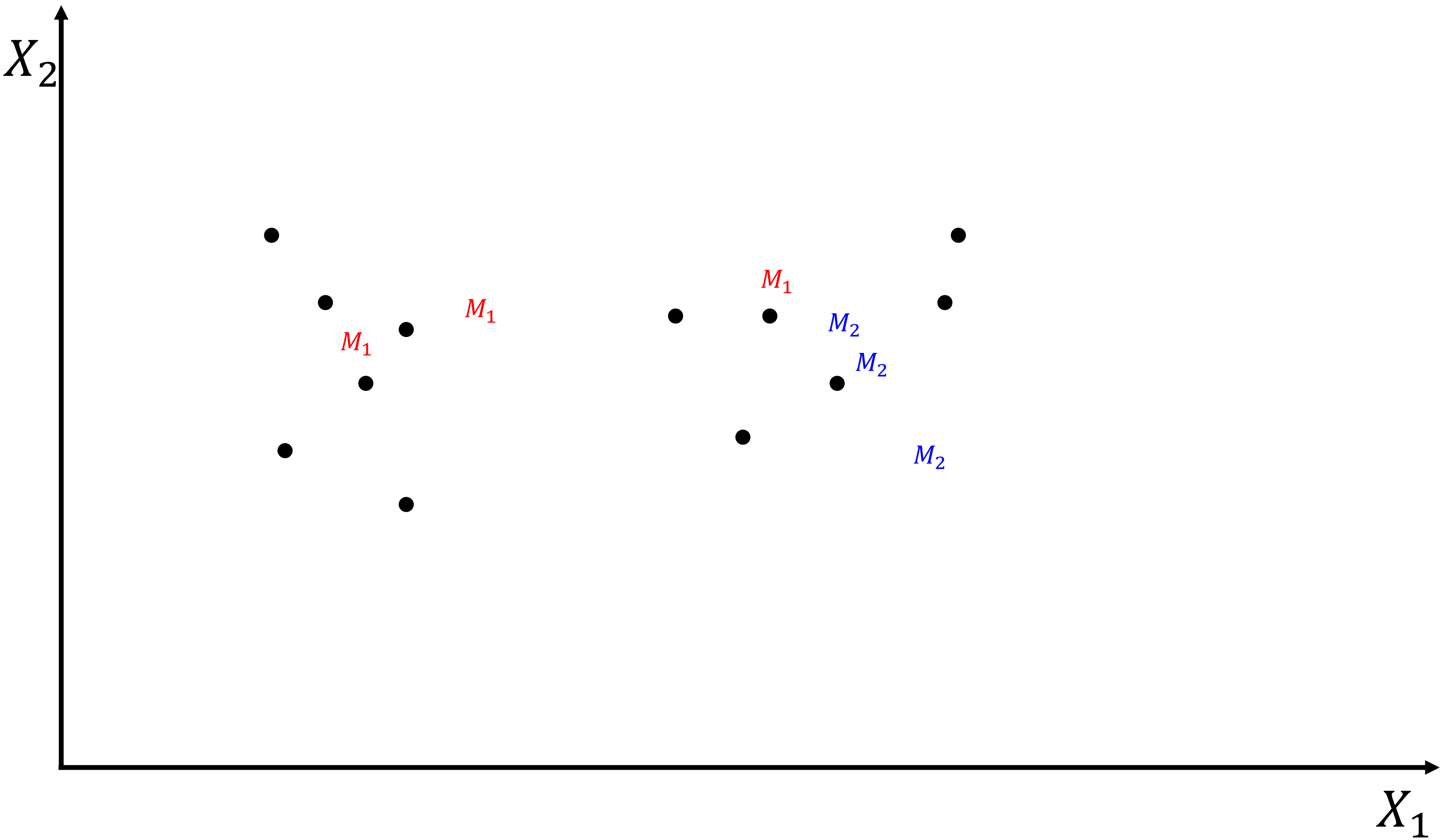
- Hyperparameter: The number of clusters k

initialize $\mu_1, \dots, \mu_k$ (randomly)
loop:
    assign all n samples to their nearest μ_i
    compute $\mu_1, \dots, \mu_k$ using their cluster members
until no change in $\mu_1, \dots, \mu_k$
return $\mu_1, \dots, \mu_k$

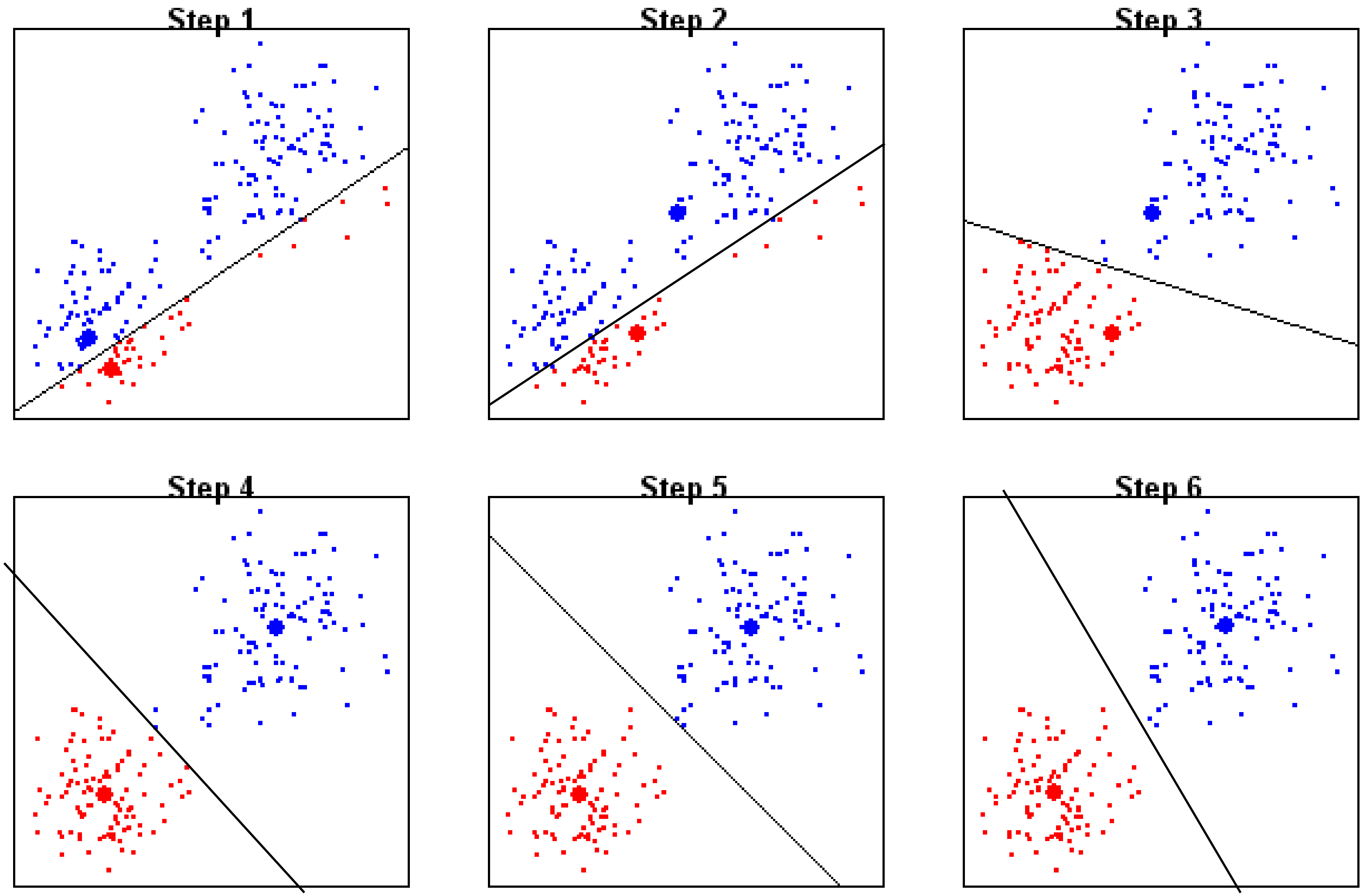
Two Inner Loops

- Assignment: loop over all instances and assign them to their closest centroid
  - Usually by checking the Euclidean distance to all centroids
  - How many distances do we have to compute?
- Recomputing centroids: loop over clusters (subsets) and calculate new centroids
  - Complexity?
  - Extension to different representative calculations?

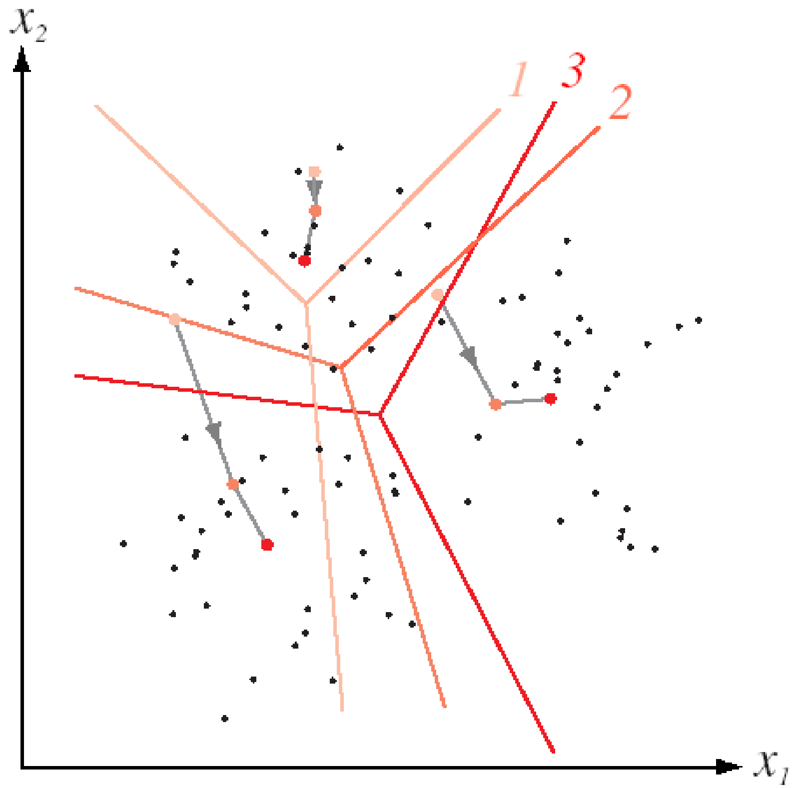
k-Means Clustering Algorithm



Boundaries Between Clusters



Boundaries are Linear – Why?



What About the Second Criteria?

- We want to **increase the compactness** of the cloud of points **within** each one of the clusters.
- We also want to **increase the distance between** different clusters.

Total Scatter and its Magic Implications

- The total variance of the set of points D (or the scatter) is defined as:

$$S_T(D) = \frac{1}{m} \sum_{x \in D} \|x - \mu\|^2$$

Where μ is the mean of all the samples in D

- The total scatter <u>does not depend</u> on the clustering!
- However - for any proposed centroid based clustering, we will show that:

$$S_T(D) = S_C\left(D, \{C_i\}_{i=1}^{k}\right) + \sum_{i=1}^{k} \frac{n_i}{m} \cdot \|\mu_i - \mu\|^2$$

Total Scatter Decomposition

- $S_T(D) = \cancel{\frac{1}{m}} \sum_{x \in D} \|x - \mu\|^2$
- $\mathrm{S_C}\left(D, \{C_i\}_{i=1}^k\right) = \cancel{\frac{1}{m}} \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2$
- $S_B\left(D, \{C_i\}_{i=1}^k\right) = \sum_{i=1}^k \frac{n_i}{\cancel{m}} \cdot \|\mu_i - \mu\|^2$

$$S_T(D) = \mathrm{S_C}\left(D, \{C_i\}_{i=1}^k\right) + \mathrm{S_B}\left(D, \{C_i\}_{i=1}^k\right)$$

Total Scatter

Within Scatter

Between Scatter

Total Scatter decomposition

$$S_T = S'_T(D) = \sum_{x \in D} \|x - \mu\|^2$$

$$S_C = S'_C\left(D, \{C_i\}_{i=1}^{k}\right) = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 = \sum_{i=1}^{k} \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^T$$

$$S_B = {S_B}'\left(D, \{C_i\}_{i=1}^{k}\right) = \sum_{i=1}^{k} n_i \|\mu_i - \mu\|^2 = \sum_{i=1}^{k} n_i (\mu_i - \mu)(\mu_i - \mu)^T$$

$$S_T = S_C + S_B$$

$$S_C + S_B = \quad S_C + \sum_{i=1}^{k} (n_i\mu_i - n_i\mu_i)(\mu_i - \mu)^T + \sum_{i=1}^{k} (\mu_i - \mu)^T (n_i\mu_i - n_i\mu_i) + S_B =$$

(Because $\sum_{x\in C_i} \mu_i = n_i\mu_i = \sum_{x\in C_i} x$)

$$= S_C + \sum_{i=1}^{k} \left(\sum_{x\in C_i} x - \sum_{x\in C_i} \mu_i \right) (\mu_i - \mu)^T + \sum_{i=1}^{k} (\mu_i - \mu) \left(\sum_{x\in C_i} x - \sum_{x\in C_i} \mu_i \right) + S_B =$$

$$= \sum_{i=1}^{k} \sum_{x\in C_i} (x - \mu_i)(x - \mu_i)^T + \sum_{i=1}^{k} \sum_{x\in C_i} (x - \mu_i)\,(\mu_i - \mu)^T$$

$$+ \sum_{i=1}^{k} \sum_{x\in C_i} (\mu_i - \mu)(x - \mu_i)^T + \sum_{i=1}^{k} \sum_{x\in C_i} (\mu_i - \mu)(\mu_i - \mu)^T =$$

$$= \sum_{i=1}^{k} \sum_{x\in C_i} \left[(x - \mu_i)(x - \mu_i)^T + (x - \mu_i)(\mu_i - \mu)^T + (\mu_i - \mu)(x - \mu_i)^T + (\mu_i - \mu)(\mu_i - \mu)^T \right] =$$

$$= \sum_{i=1}^{k} \sum_{x\in C_i} \left[(x - \mu_i) + (\mu_i - \mu) \right] \left[(x - \mu_i) + (\mu_i - \mu) \right]^T =$$

$$= \sum_{i=1}^{k} \sum_{x\in C_i} (x - \mu)(x - \mu)^T = \sum_{x\in D} (x - \mu)(x - \mu)^T = S_T$$

Total Scatter and its Magic Implications

- The total scatter does not depend on the clustering!
- For any proposed centroid based clustering, we have:

$$S_T(D) = \mathrm{S_C}\left(D, \{C_i\}_{i=1}^{k}\right) + \mathrm{S_B}\left(D, \{C_i\}_{i=1}^{k}\right)$$

- There is a clear monotone tradeoff between the two scatter terms: When one goes up the other goes down.
- This means that minimizing within cluster scatter will **also** maximize between cluster scatter

Total Scatter and its Magic Implications

$$S_T(D) = \mathrm{S_C}\left(D, \{C_i\}_{i=1}^k\right) + \mathrm{S_B}\left(D, \{C_i\}_{i=1}^k\right)$$

- Minimizing within cluster scatter:

$$\mathrm{S_C}\left(D, \{C_i\}_{i=1}^k\right) = \frac{1}{m}\sum_{i=1}^{k}\sum_{x \in C_i} \|x - \mu_i\|^2$$

- Will also maximize between cluster scatter:

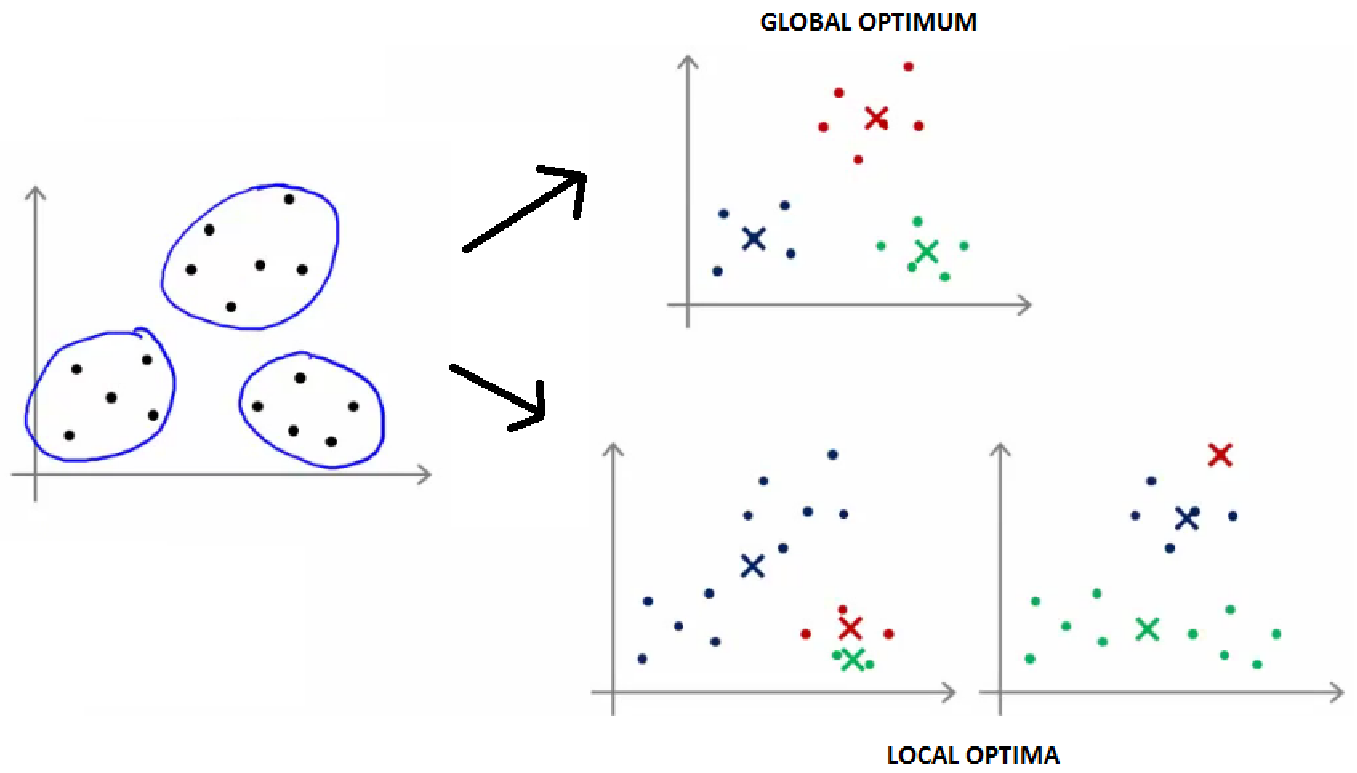
$$S_B\left(D, \{C_i\}_{i=1}^k\right) = \sum_{i=1}^{k}\frac{n_i}{m} \cdot \|\mu_i - \mu\|^2$$

- We therefore have an adequate representation of the total error of representing the m samples by a set of k specified clusters.

Guaranteed Convergence

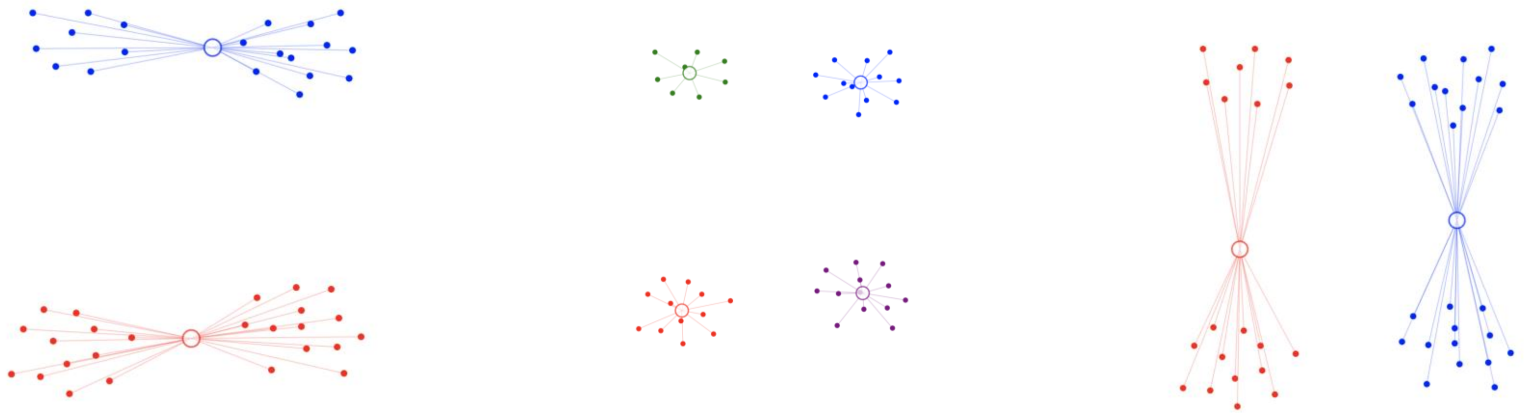
- At each iteration, $\frac{1}{m}\sum_{i=1}^{k}\sum_{x\in C_i}||x-\mu_i||^2$ is (weakly) reduced:
  - Assignment: if an instance is closer to a different representative, then it is re-assigned and the function is reduced
  - Recomputing representatives: the centroid of a subset minimizes the average distances of all the set
- There are only a finite number of possible assignments (although very large) so the function is bounded from below (minimum of all possible assignments)
- Therefore, it must converge!
- However, it may converge to a local minimum…

Local Minima Example



Demo

- http://user.ceng.metu.edu.tr/~akifakkus/courses/ceng574/k-means/



Heuristic Enhancements

- Run k-means several times with different random initializations and choose the best local minimum

- When a representative does not represent any instance…
  - Discard it (revert to k-1 means clustering)
  - Choose a new random representative
  - Choose the largest error cluster and split it to two

k-Means Image Compression

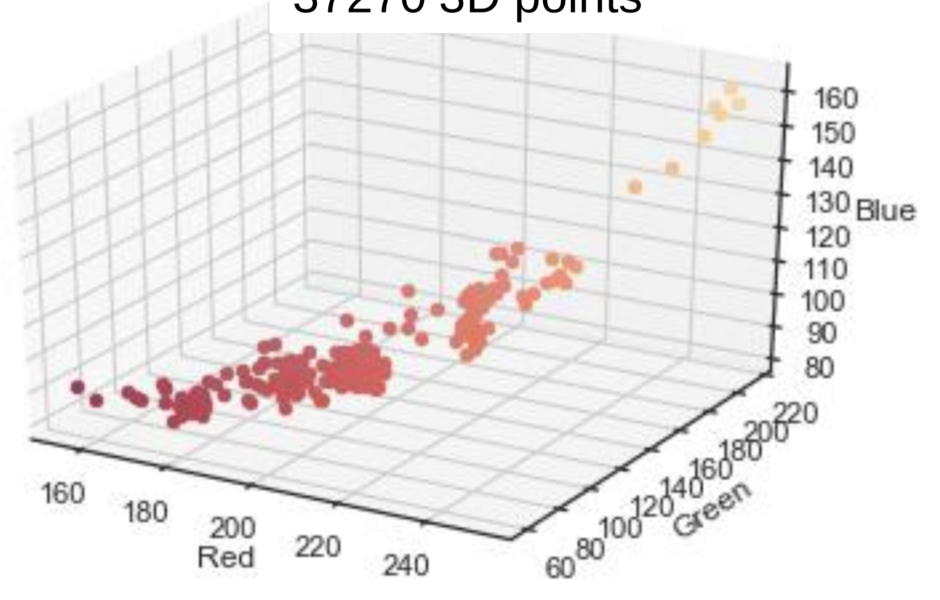
- Each pixel has 3 values: R, G, B
- We have a 3D data-set with HxW colors (worst case)
- We want to assign only K colors – how?

37270 Colors

Clustering Colors

- We search for K best representative colors
- Use k-Means algorithm

37270 3D points



Choosing K?

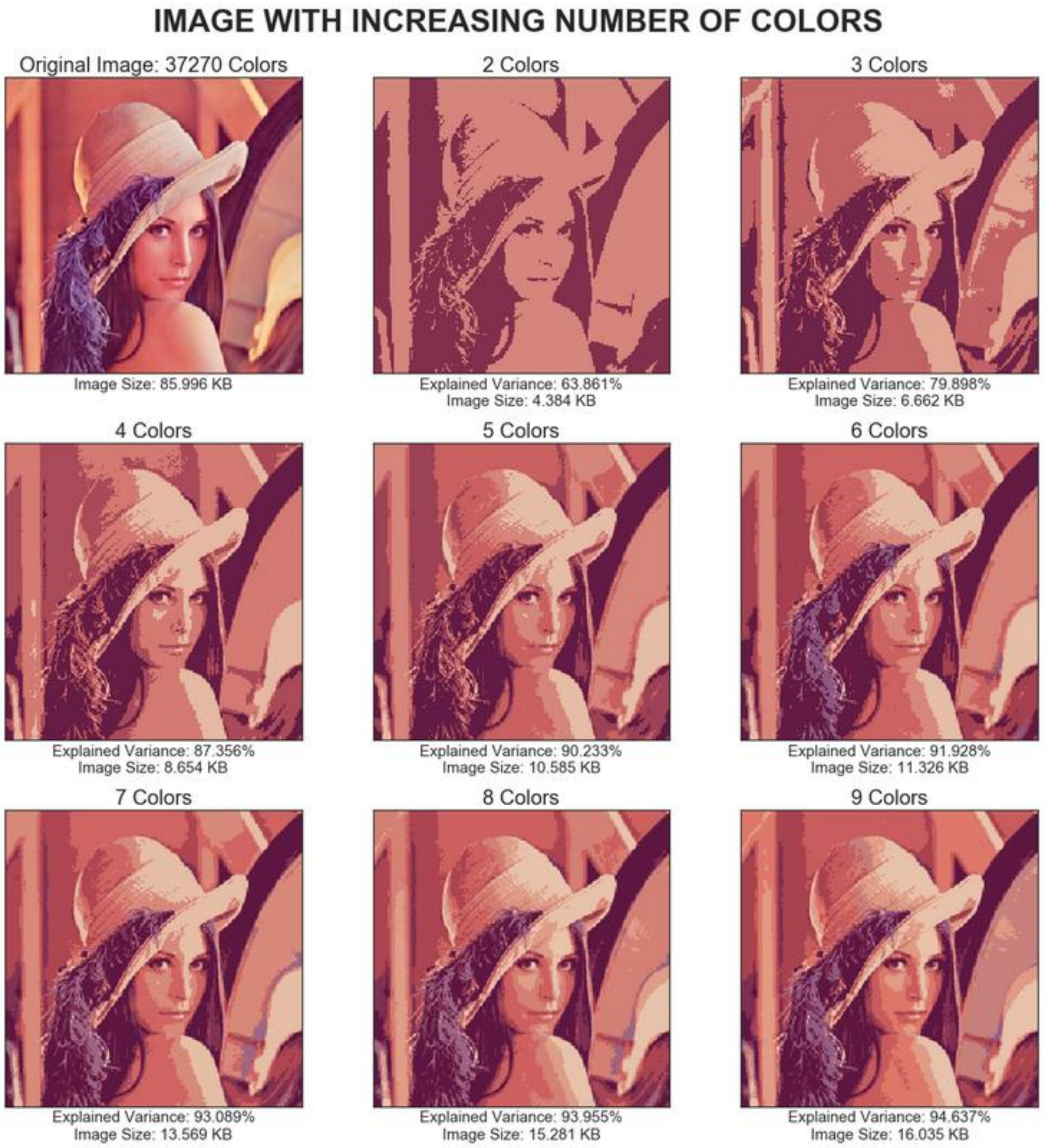
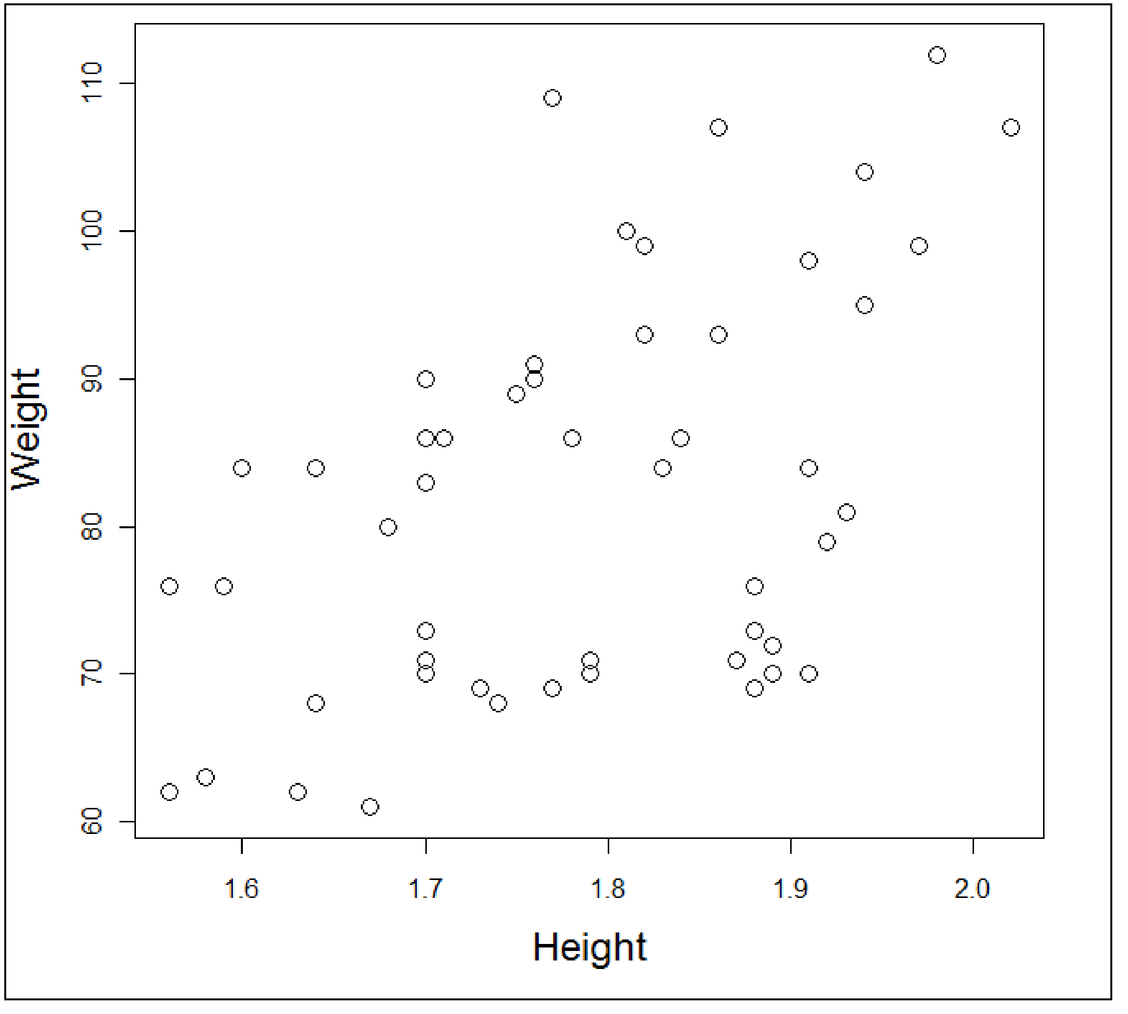


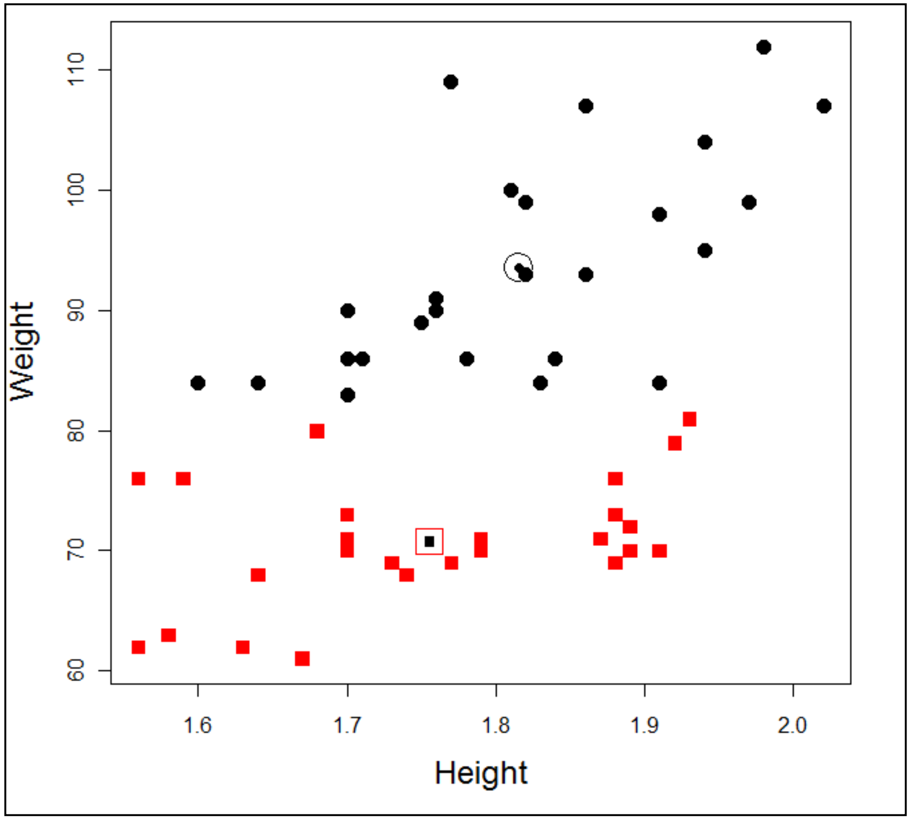
IMAGE WITH INCREASING NUMBER OF COLORS
Original Image: 37270 Colors
Image Size: 85.996 KB
2 Colors
Explained Variance: 63.861%
Image Size: 4.384 KB
3 Colors
Explained Variance: 79.898%
Image Size: 6.662 KB
4 Colors
Explained Variance: 87.356%
Image Size: 8.654 KB
5 Colors
Explained Variance: 90.233%
Image Size: 10.585 KB
6 Colors
Explained Variance: 91.928%
Image Size: 11.326 KB
7 Colors
Explained Variance: 93.089%
Image Size: 13.569 KB
8 Colors
Explained Variance: 93.955%
Image Size: 15.281 KB
9 Colors
Explained Variance: 94.637%
Image Size: 16.035 KB

Example: Choosing k

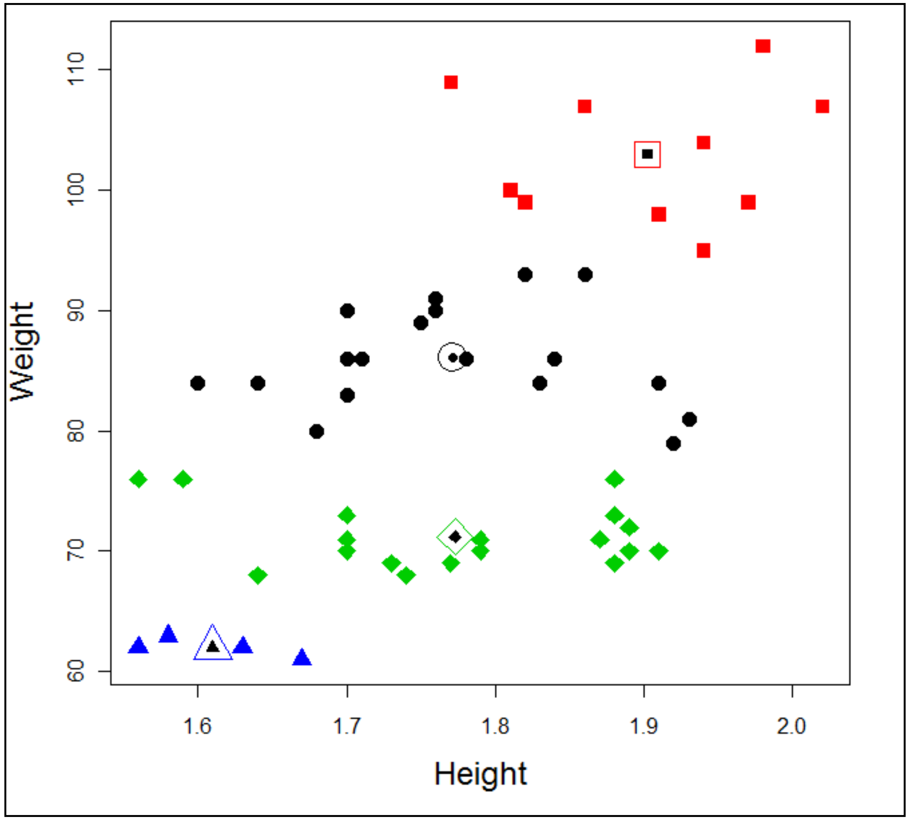


Example: Choosing k

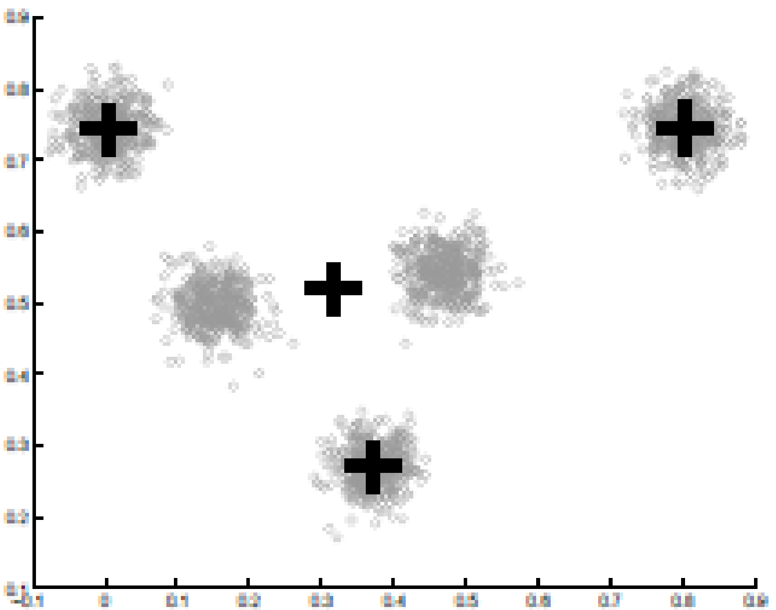
$k = 2$

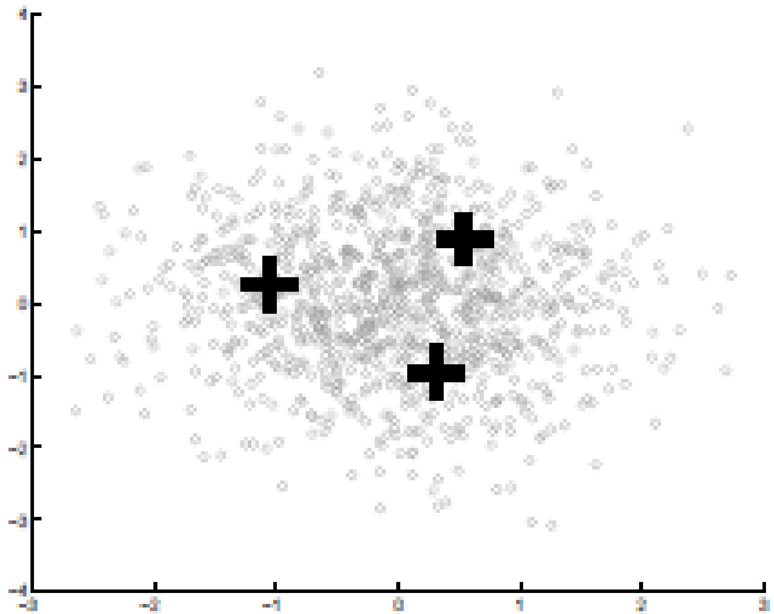


$k = 4$



Choosing the Wrong k





Choosing k

- No complete/theoretically-justified answer to this question
- Many times, it depends on the data or your business goal (e.g. how many different types you want to support in your product)
- The larger the k, the smaller the actual minimal value of

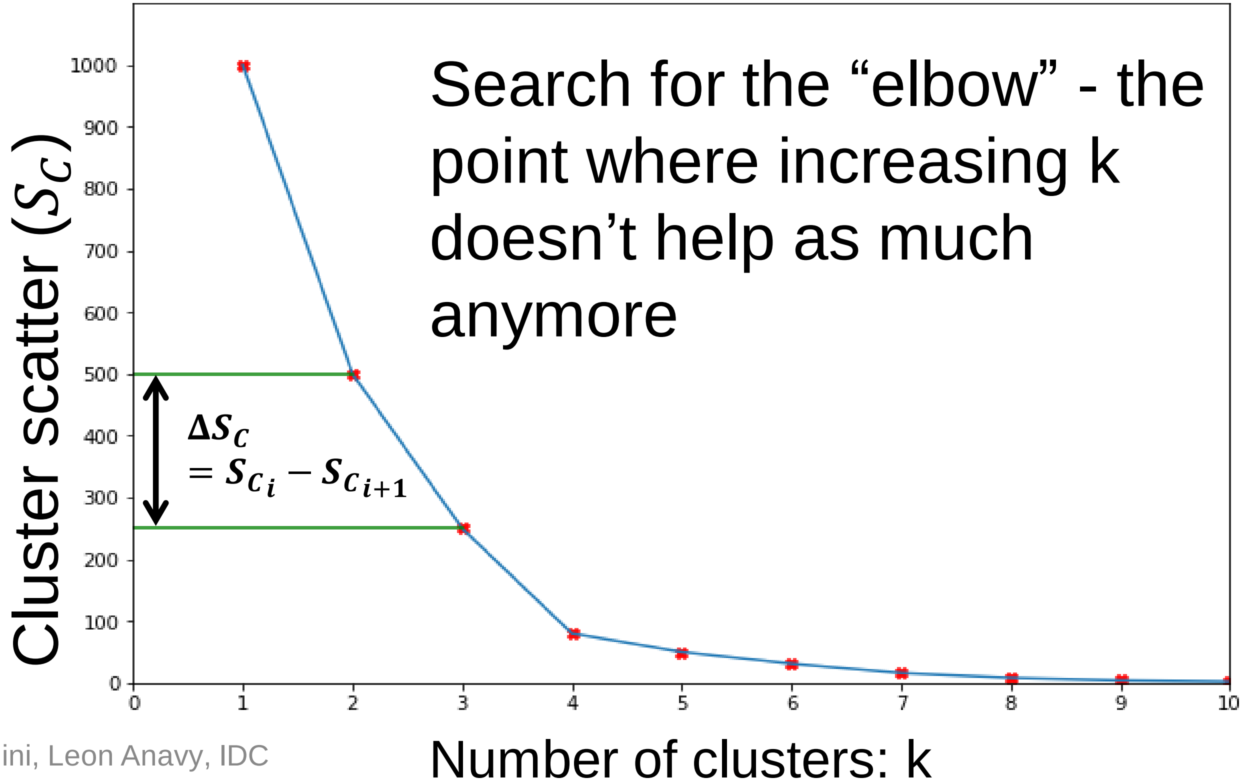
$$S_C\left(D, \{C_i\}_{i=1}^{k}\right) = \frac{1}{m}\sum_{i=1}^{k}\sum_{x \in C_i} \|x - \mu_i\|^2$$

Why?

There is no penalty in S_C for increasing k

Choosing k

- The larger the k, the smaller the actual minimal value of S_C
- Note - optimal value for $S_C(k) < S_C(k+1)$ (Why?)
- When $k = m$ we get $S_C = 0$



Silhouette Measure

- Measuring how “comfortable” each sample is in its cluster
- Define for each sample $x \in C_i$:
  - Average distance to its “buddies” (in the same cluster)

$$a(x) = \frac{\sum_{y \in C_i} d(x, y)}{|C_i|}$$

  - Average distance to the nearest cluster (to which it’s not assigned)

$$b(x) = \min_{C_j \neq C_i} d\left(x, C_j\right)$$

$$d\left(x, C_j\right) = \frac{\sum_{y \in C_j} d(x, y)}{\left|C_j\right|}$$

Silhouette Measure

$$S(x) = \frac{b(x) - a(x)}{\max\{b(x), a(x)\}} \in [-1,1]$$

$$S(x) \approx \begin{cases} 1 & x\ is\ well\ classified\ in\ cluster\ C_i \\ 0 & x\ lies\ on\ the\ border\ between\ C_i\ and\ C_j \\ -1 & x\ is\ badly\ classified\ in\ cluster\ C_i\ (closer\ to\ C_j) \end{cases}$$

Silhouette Measure

- Silhouette Coefficient (SC) of the clustering result

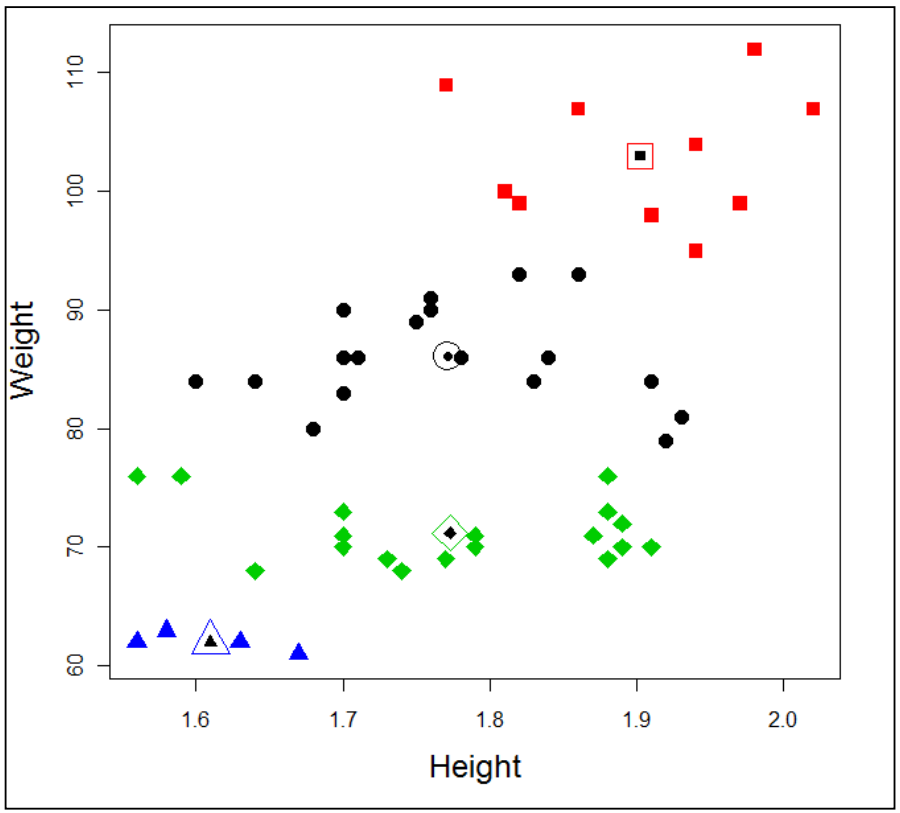
$$SC = \frac{1}{m} \sum_{x \in D} S(x)$$

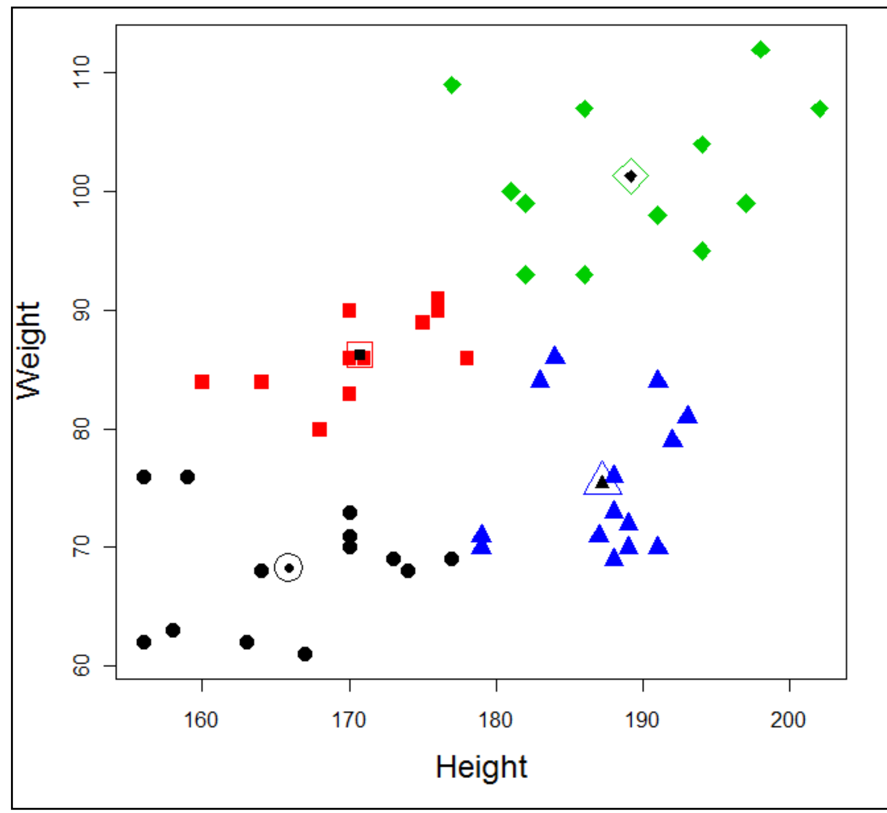
- Rule of thumb:

$$\begin{array}{ccl} 0.7 \le SC & - & strong\ cluster\ structure \\ 0.5 \le SC \le 0.7 & - & reasonable\ structure \\ 0.25 \le SC \le 0.5 & - & weak\ structure \\ SC \le 0.25 & - & no\ structure \end{array}$$

Sensitivity to Data Units

Height [m] → Height [cm]





Fuzzy (soft) k-means

- Instead of assigning each instance to a cluster, we use a measure of “how close” it is to each cluster.

  For example, one can use

$$e^{-\|x-\mu_i\|^2}$$

- Later this can be interpreted as the probability that an instance belongs to each cluster

Fuzzy (soft) k-means

- Hyperparameter: The number of clusters k

initialize $\mu_1, \ldots, \mu_k$ (randomly)
loop:
    Calculate fuzzy assignments of all samples x to all μ_i using $e^{-\|x-\mu_i\|^2}$
    compute $\mu_1, \ldots, \mu_k$ using weighted averages
until no change in $\mu_1, \ldots, \mu_k$
return $\mu_1, \ldots, \mu_k$

Fuzzy k-means vs. Standard k-mean

- Assignment step: fuzzy membership vs. binary membership
- For each instance x defines a vector of dimension k:

$$v = \left(e^{-\|x-\mu_1\|^2}, e^{-\|x-\mu_2\|^2}, \dots, e^{-\|x-\mu_k\|^2}\right)$$

- Optional: normalize $v = \frac{v}{\|v\|}$
- Centroid calculation step: weighted mean (weighted by the probability) vs. simple mean:

$$\mu_i = \frac{\sum_{x \in D} v_i x}{\sum_{x \in D} v_i}, \qquad i = 1, \dots, k$$

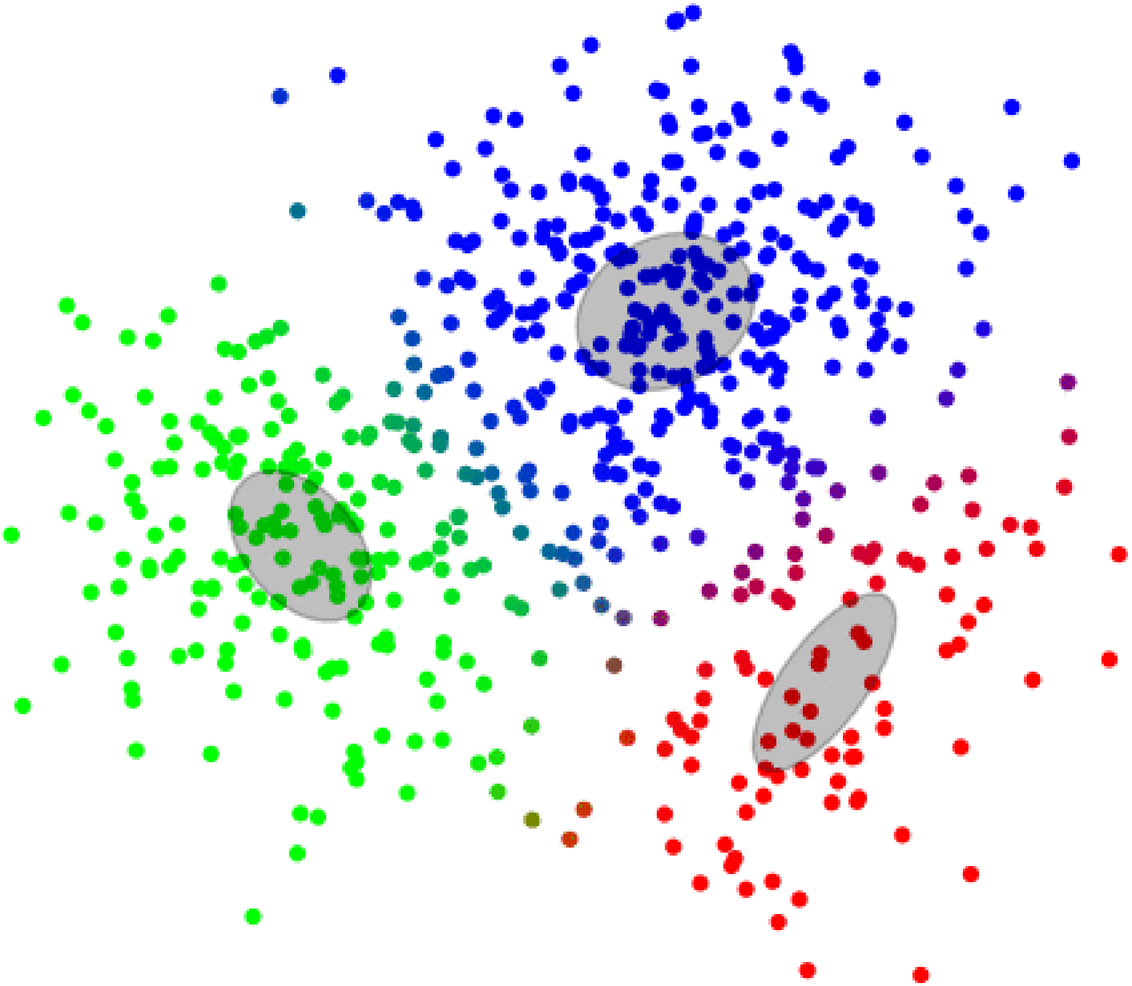
- For a rigid assignment of an instance x we can use the cluster

$$\mathrm{j}^* = \operatorname*{argmax}_j v_j$$

Probabilistic View of Fuzzy k-means

- If we normalized the distances of each instance, then we can interpret v_i as the probability of x to be assigned to cluster j
- Next, we can ask about the probability of each cluster given the data, i.e.
  - $v_j = P(\mathrm{C} = C_j | X = x)$
- We do not know these posterior probabilities what can we do?
- Bayes: $P\left(C_j \middle| x\right) = \frac{P(x|C_j)P(C_j)}{P(x)}$

Fuzzy Clusters



K-means - Summary

- Key factors in clustering:
  - Similarity measures
  - Quality criteria functions (evaluation)
  - Algorithm (optimization of criteria)
- Efficient and simple clustering algorithm: k-means
  - Simultaneously min and max of the respective scatters
  - Convergence guaranteed
  - How to determine k?
  - Variants: fuzzy k-means, k-medoids, others

Other Possible Clustering Algorithms

- Sometimes we want to work “model free” and not set k in advance
- We want to “let the data speak” during the clustering algorithm.
- Observing the complete structure of the data and not just the final clustering result

Hierarchical Clustering

- Bottom-up approach – agglomerative clustering
  - Start with every sample in its own cluster
  - In every step combine the two closest clusters

- Top-down approach – Divisive clustering
  - Start with all the samples in the same cluster
  - In every step split the least "compact" cluster to two

Bottom-up - Agglomerative Clustering

start with every sample in a cluster

loop:

calculate the distance matrix between
all the clusters

merge the two closest clusters

update the matrix

until the is only one cluster co ing all the samples

Need to define a distance measure between clusters

Calculating Distance Between Clusters

- Average distance:

$$D(R,Q) = \frac{1}{n_R n_Q} \sum_{x \in R, y \in Q} d(x,y)$$

- Nearest neighbor (Single Linkage):

$$D(R,Q) = \min_{x \in R, y \in Q} d(x,y)$$

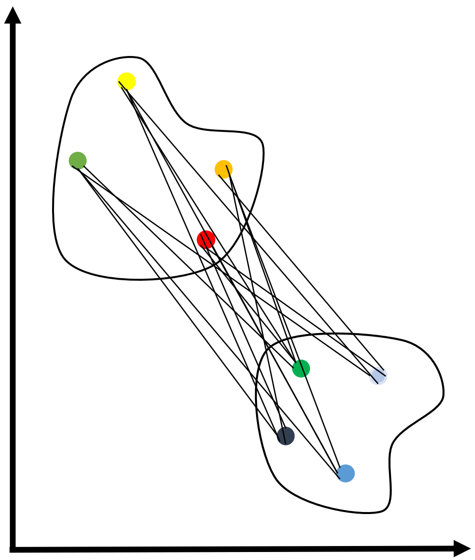
- Maximal distance (Complete Linkage):

$$D(R,Q) = \max_{x \in R, y \in Q} d(x,y)$$

- Means distance:

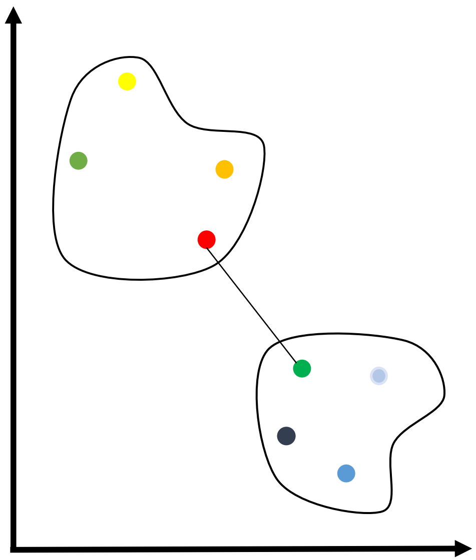
$$D(R,Q) = \left|\mu_R - \mu_Q\right|$$

Calculating distance between clusters



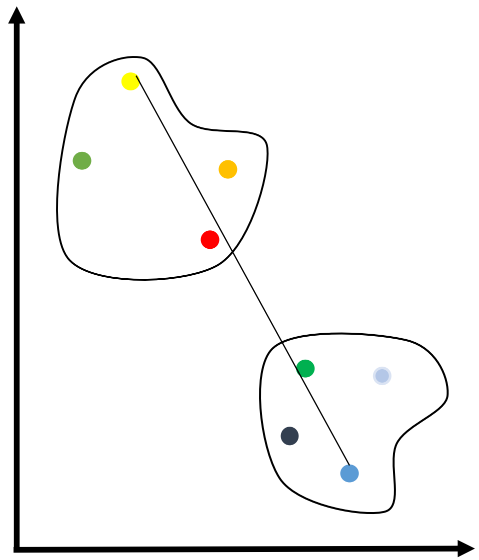
Average dissimilarity

$$D(R,Q) = \frac{1}{n_R n_Q} \sum_{x \in R, y \in Q} d(x,y)$$



Single linkage

$$D(R,Q) = \min_{x \in R, y \in Q} d(x,y)$$



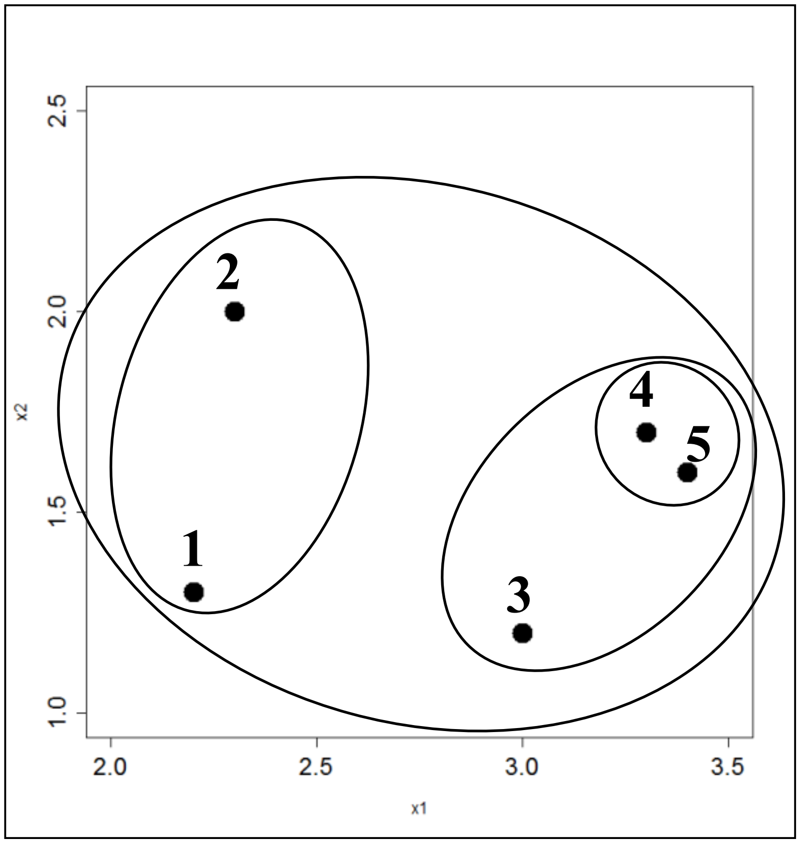
Complete linkage

$$D(R,Q) = \max_{x \in R, y \in Q} d(x,y)$$

Bottom Up Agglomerative Clustering

Nearest neighbor (Single Linkage):

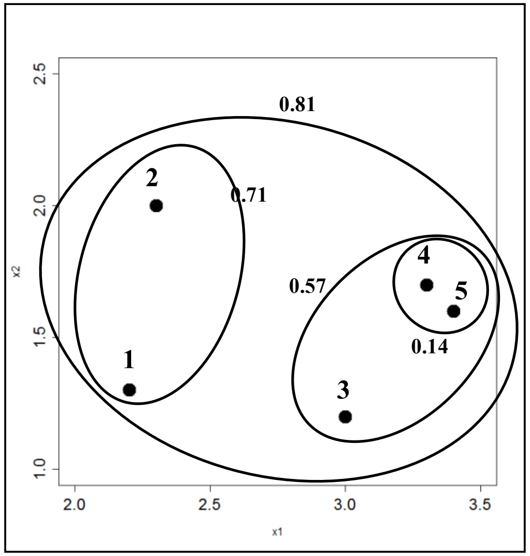
$$D(R,Q) = \min_{x \in R, y \in Q} d(x,y)$$



Dissimilarities :

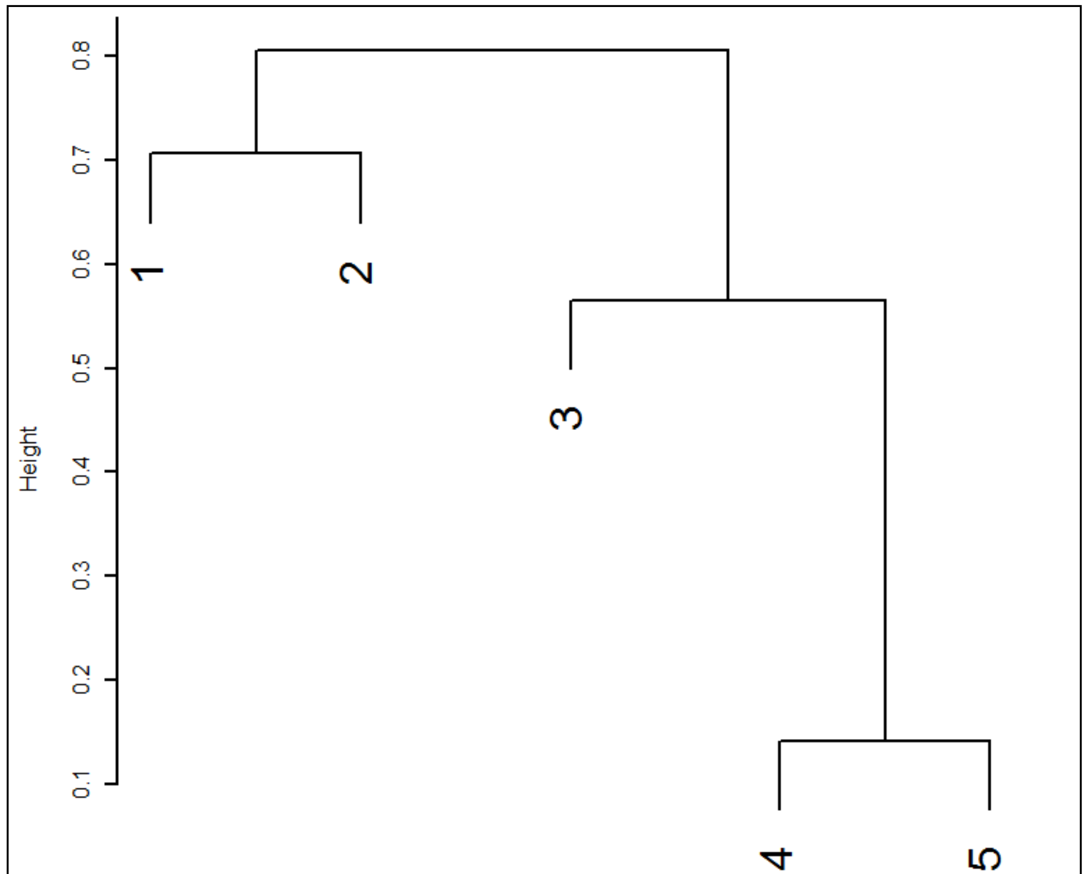
	1	2	3	4
2	~~0.71~~			
3	0.81	~~1.06~~		
4	~~1.17~~	~~1.04~~	~~0.58~~	
5	~~1.24~~	~~1.17~~	~~0.57~~	~~0.14~~

Dendrogram

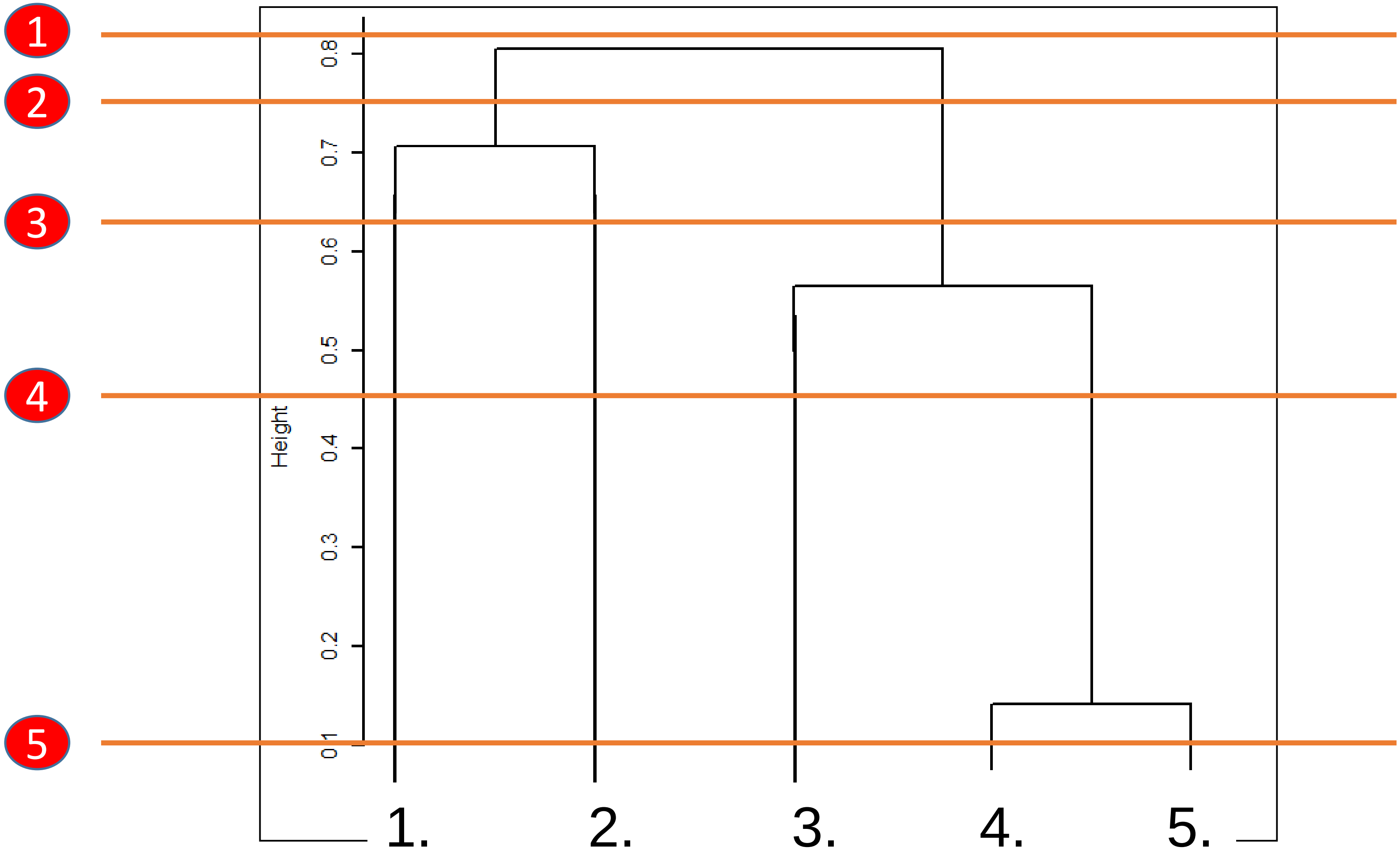


Dissimilarities :

	1	2	3	4
2	0.71			
3	0.81	1.06		
4	1.17	1.04	0.58	
5	1.24	1.17	0.57	0.14



Choosing the Number of Clusters?



Dissimilarity Measure

- If we just have a measure of dissimilarity $\delta(x,y)$ for every pair of samples where:
  - $\delta(x,y) \geq 0$ and
  - $\delta(x,y)=0$ iff $x=y$
- We can still use the bottom up approach using one of the measures:

$$\delta_{\min}(D_i, D_j) = \min_{\substack{\mathbf{x}\in D_i \\ \mathbf{y}\in D_j}} \delta(\mathbf{x},\mathbf{y})$$

$$\delta_{\max}(D_i, D_j) = \max_{\substack{\mathbf{x}\in D_i \\ \mathbf{y}\in D_j}} \delta(\mathbf{x},\mathbf{y})$$

Induced Metrics

- After hierarchical clustering we will get an induced metric between samples
- The distance d(x,y) will be the lowest level in the hierarchy for which x and y are in the same cluster.
- This measure satisfies the requirement for a metric!

Clustering - Summary

- Similarity measures
- Quality criteria functions (evaluation)
- Clustering as optimization
- Algorithms:
  - Naïve growing
  - K-means
  - Heirarchical
- Model based vs. no model
- Complete structure vs. final clustering