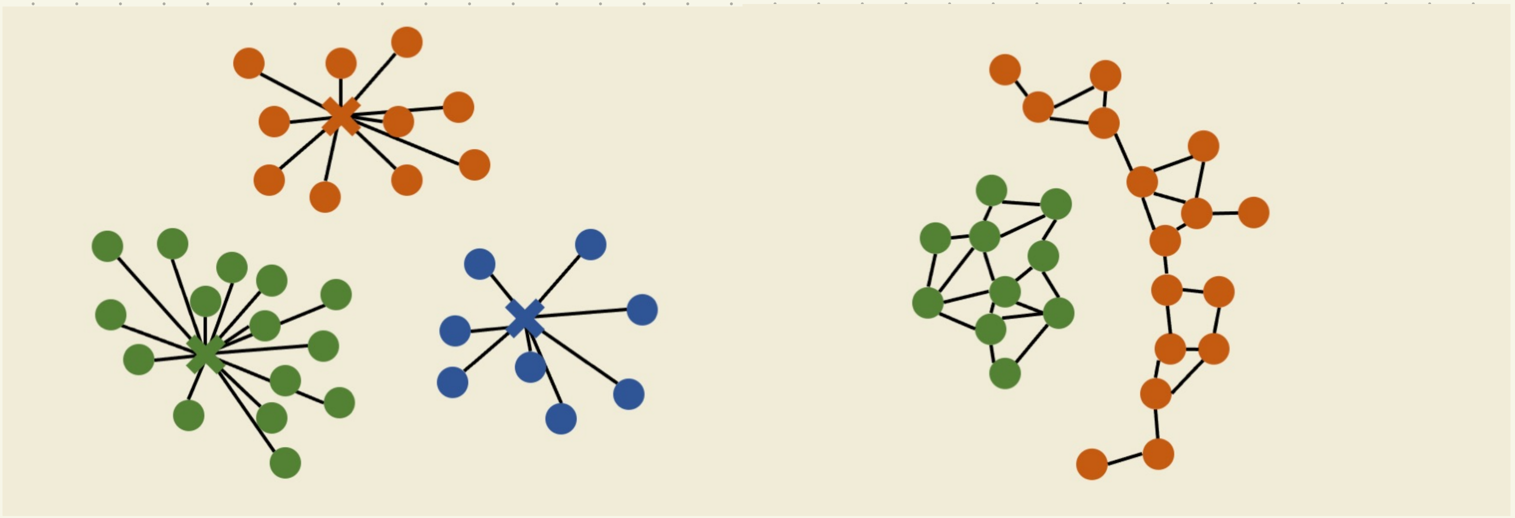
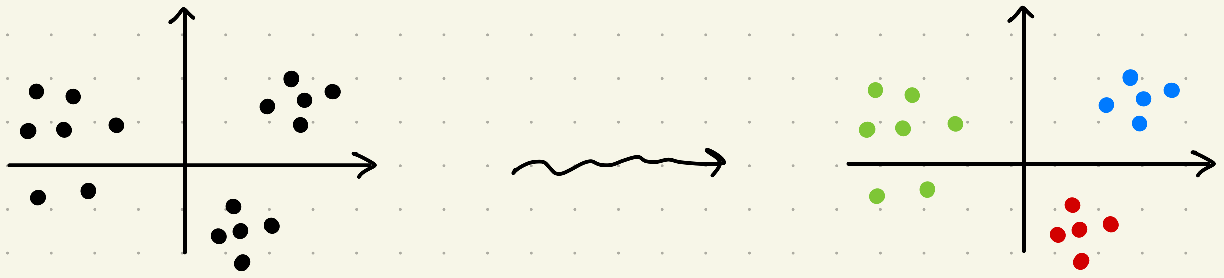
Clustering



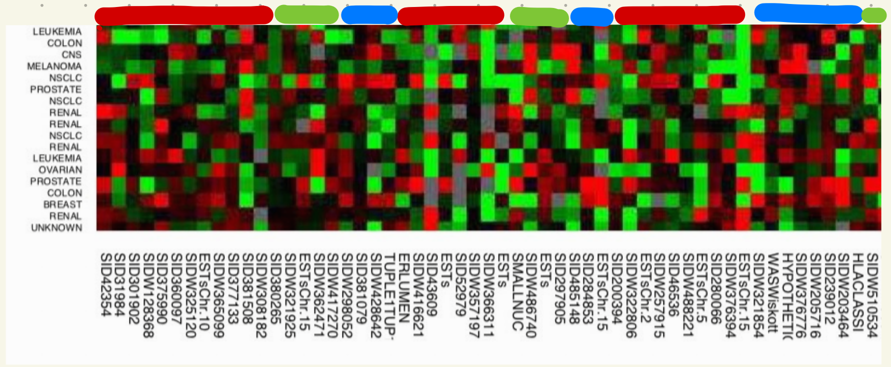
https://www.blopig.com/blog/2023/04/an-overview-of-clustering-algorithms/

Clustering

Split data into groups of similar points.



Example: group genes displaying simalarities.



K-means clustering

Given $x_1, \dots x_n \in \mathbb{R}^p$ assign them to K clusters.

$$\begin{array}{cccccc} x_1 & x_2 & \cdots\cdots & x_{n-1} & x_n \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ 3 & 2 & & 2 & K \end{array}$$

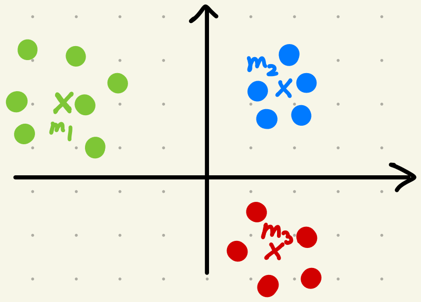
Assume that points in the same cluster are close is Euclidean distance.

<u>Goal</u>: find $m_1 \dots m_K \in \mathbb{R}^p$ cluster centroids

and $z_1 \dots z_n \in \{1 \dots K\}$ cluster assignment

mimimizing the distortion function

$$J(\underbrace{\{z_i\}_{i=1}^n}_{z}, \underbrace{\{m_k\}_{k=1}^K}_{m}) = \sum_{i=1}^n \| x_i - m_{z_i} \|^2$$



Algorithm

Input: points $x_1 \dots x_n \in \mathbb{R}^p$, number of clusters K

arbitrary centroids $m_1 \dots m_K \in \mathbb{R}^p$

Step 1: Assign each point to the closest centroid

$$z_i = \underset{k=1 \dots K}{\operatorname{argmin}} \|x_i - m_k\|^2 \quad \text{for} \quad i = 1 \dots n$$

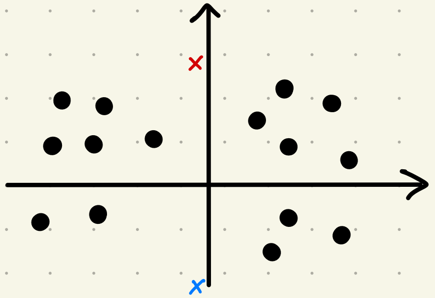
Step 2: Update centroids with cluster means

$$m_k = \frac{\sum_{i=1}^{n} I(z_i = k)\, x_i}{\sum_{i=1}^{n} I(z_i = k)} \quad \text{for} \quad k = 1 \dots K$$

Output: cluster assignments $z_1 \dots z_n$

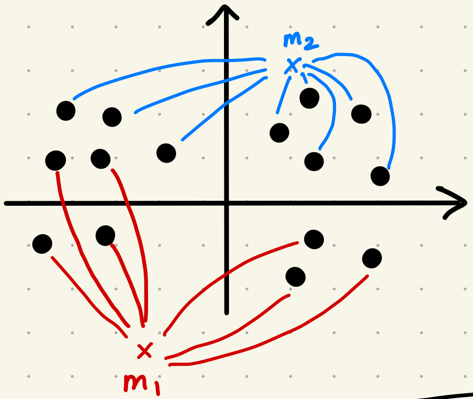
cluster centers $m_1 \dots m_K$.

Input
(K=2)

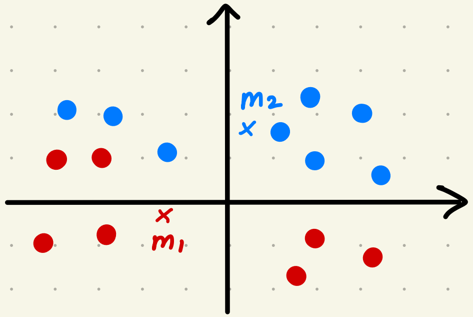


Iteration 1

Step 1

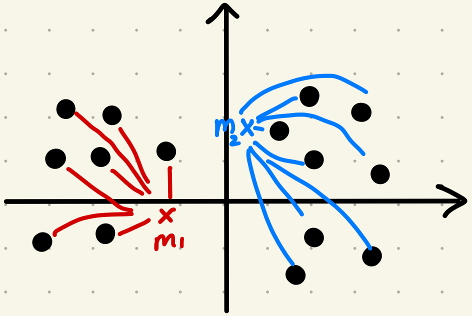


Step 2

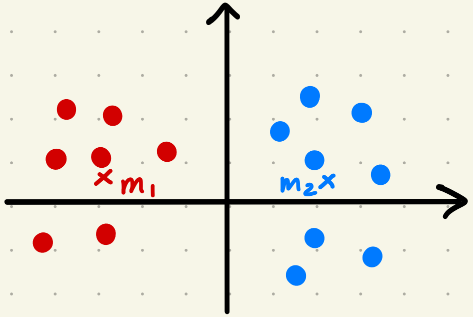


Iteration 2

Step 1



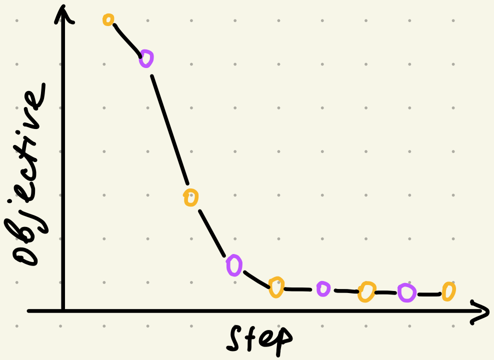
Step 2



You can show that

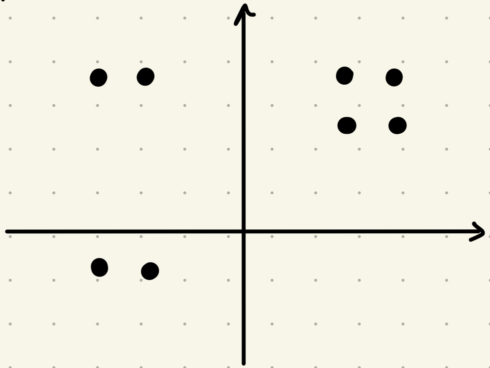
- Step 1 decreases the objective $J(z,m)$
- Step 2 decreases the objective $J(z,m)$

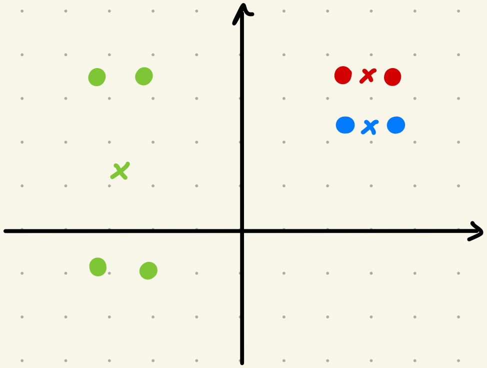
So the objective $J(z,m)$ will converge (HW3)



It is possible to get stuck at local minima

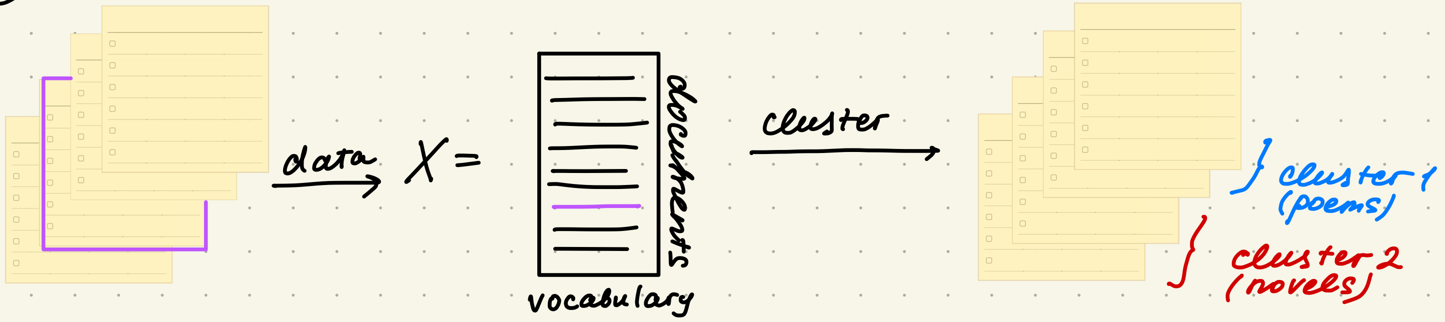
Input
($K=3$)





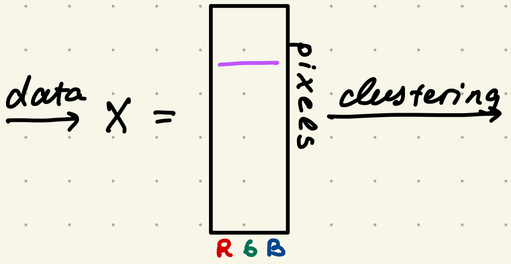
Applications

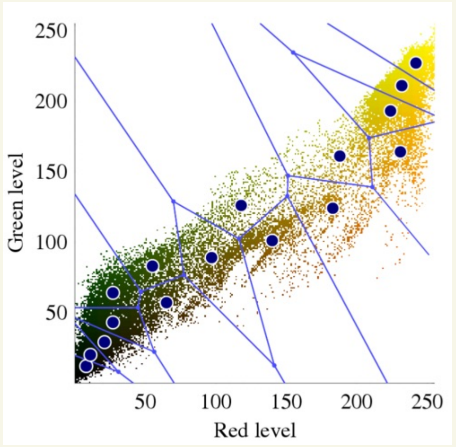
① Documentation classification



② Image compression (vector quantization)

reduce color pallette to K colors





Limitations:

- k-means is sensitive to outliers
- Needs random restarts.

Modifications:

① Replace distances $\|x-y\|^2$ with $d(x,y)$

$$J(z,m) = \sum_{i=1}^{n} d(x_i, m_{z_i})$$

+ May solve the outliers problem

- Step 2 may become very expensive

② k-medoids: pick centroids among the data

i.e. $m_j \in \{x_1 \dots x_n\}$.

+: we don't need to recompute $\|x_i - m_k\|^2$

instead we just store $\|x_i - x_j\|^2$.

- Step 2 is more expensive now.

③ k-means++ (Arthur and Vassilvitskii, 2008):

improves initialization

Init: pick m_1 at random from $x_1 \dots x_n$

for $j = 2 \dots k$

Step 1: compute $D^2(x_i) = \|x_i - m_{k-1}\|^2$

Step 2: Choose $m_k = x_i$ with probability proportional to $D^2(x_i)$

After selecting the centroids $m_1 \dots m_K$ run k-means.

+ : more accurate and faster than k-means

④ If we denote $w_{ik} = I(z_i = k)$ in k-means then:

$$J(z, m) = \sum_{i=1}^{n} \|x_i - m_{z_i}\|_2^2$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} w_{ik} \|x_i - m_k\|^2 = J(w, m)$$

<u>Step2</u>: Update centroids with cluster means

$$m_k = \frac{\sum_{i=1}^{n} I(z_i = k)\, x_i}{\sum_{i=1}^{n} I(z_i = k)} = \frac{\sum_{i=1}^{n} w_{ik}\, x_i}{\sum_{i=1}^{n} w_{ik}} \quad \text{for } k = 1 \ldots\ldots K$$

Soft-clustering: uses w_{ij} "probability weights" to assign observations to clusters.

minimize $J(w, m) = \sum_{i=1}^{n} \sum_{k=1}^{K} w_{ik} \|x_i - m_k\|^2$

$$0 \le w_{ik} \le 1 \;\; \forall\, i, k \;; \quad \sum_{i=1}^{n} w_{ik} > 0 \;; \quad \sum_{k=1}^{K} w_{ik} = 1$$

Choosing K

Determining K is a hard problem!

Denote by $C_1 \dots C_K \subseteq \{1, \dots, n\}$ the cluster partitioning. Thus $C_k \cap C_{k'} = \emptyset$ for $k \neq k'$ and $C_1 \cup \dots \cup C_K = \{1 \dots n\}$. Denote $n_k = |C_k|$

Example

$$\begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 2 & 1 & 1 & 2 \end{array}$$

$C_1 = \{1, 3, 4\}$ $C_2 = \{2, 5\}$ $n_1 = 3$ $n_2 = 2$

Denote by $\bar{x}_k = \frac{1}{n_k} \sum_{i \in C_k} x_i$ the cluster centers.

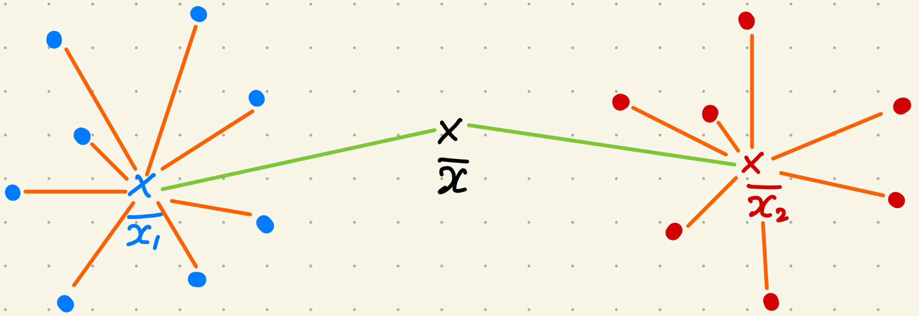
and by $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x$ the mean value across all data points.

Within-cluster scatter (variation) is

$W(k) = \sum_{k=1}^{K} \sum_{i \in C_k} \| x_i - \bar{x}_k \|^2$ – measures how tight clusters are

Between cluster scatter (variation) is

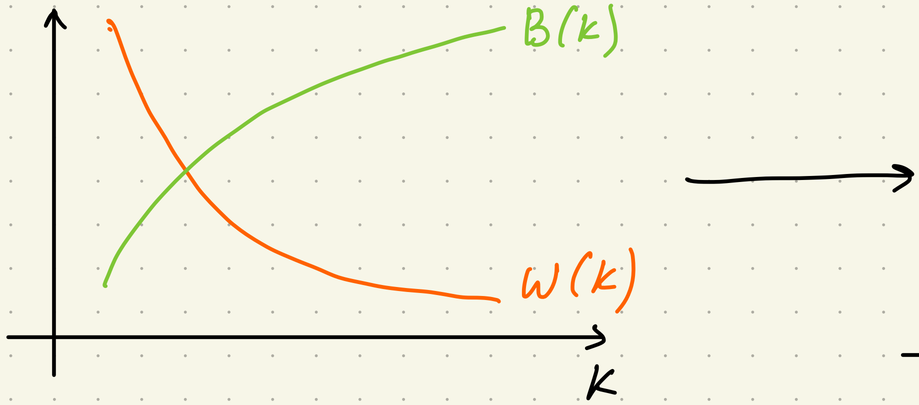
$B(k) = \sum_{k=1}^{K} n_k \| \bar{x}_k - \bar{x} \|^2$ – measures how spead apart clusters are

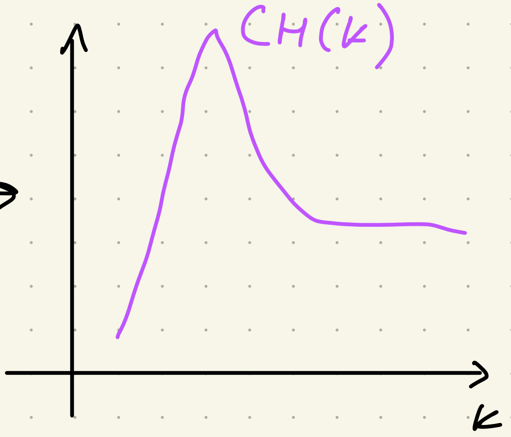


CH score (Calinski and Harabasz, 1974)

Low values of $W(k)$ are good.

High values of $B(k)$ are good.





- We can use elbow detection again.
- Alternatively, $CH(k) = \dfrac{B(k) / (k-1)}{W(k) / (n-k)}$

Pick k that has higest CH score.

Gap statistic (Tibshirani et al., 2001)

- CH cannot be computed for $k=1$.
- Gap statistic uses $W(k)$ and compares it to $W_{unif}(k)$ that is within cluster scatter computed for (simulated) uniform data.

$$Gap(k) = \underbrace{\log W_{unif}(k)}_{\text{averaged across several simulations}} - \log W(k)$$

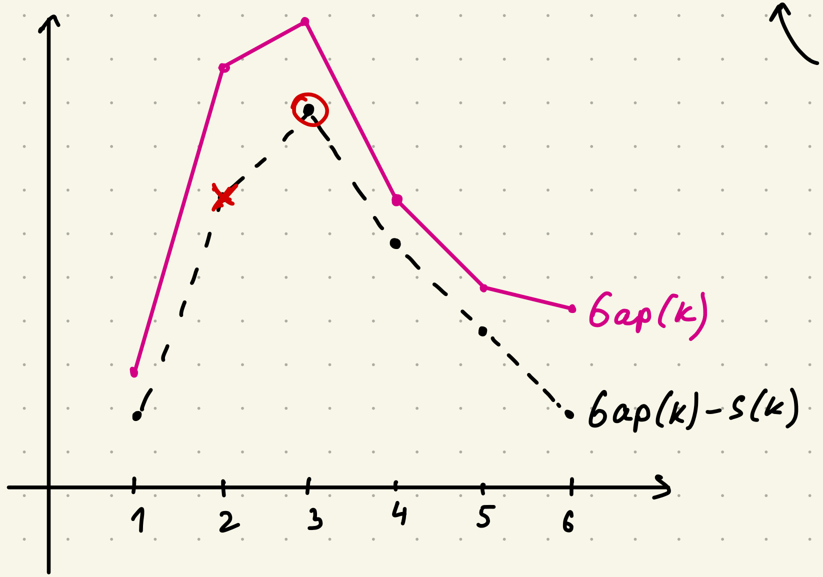
- Larger Gap means more deviation from the uniform distribution.

We also compute $S(k)$, the standard error of $\log W_{unif}(k)$ over simulations.

- Find minimum k such that:

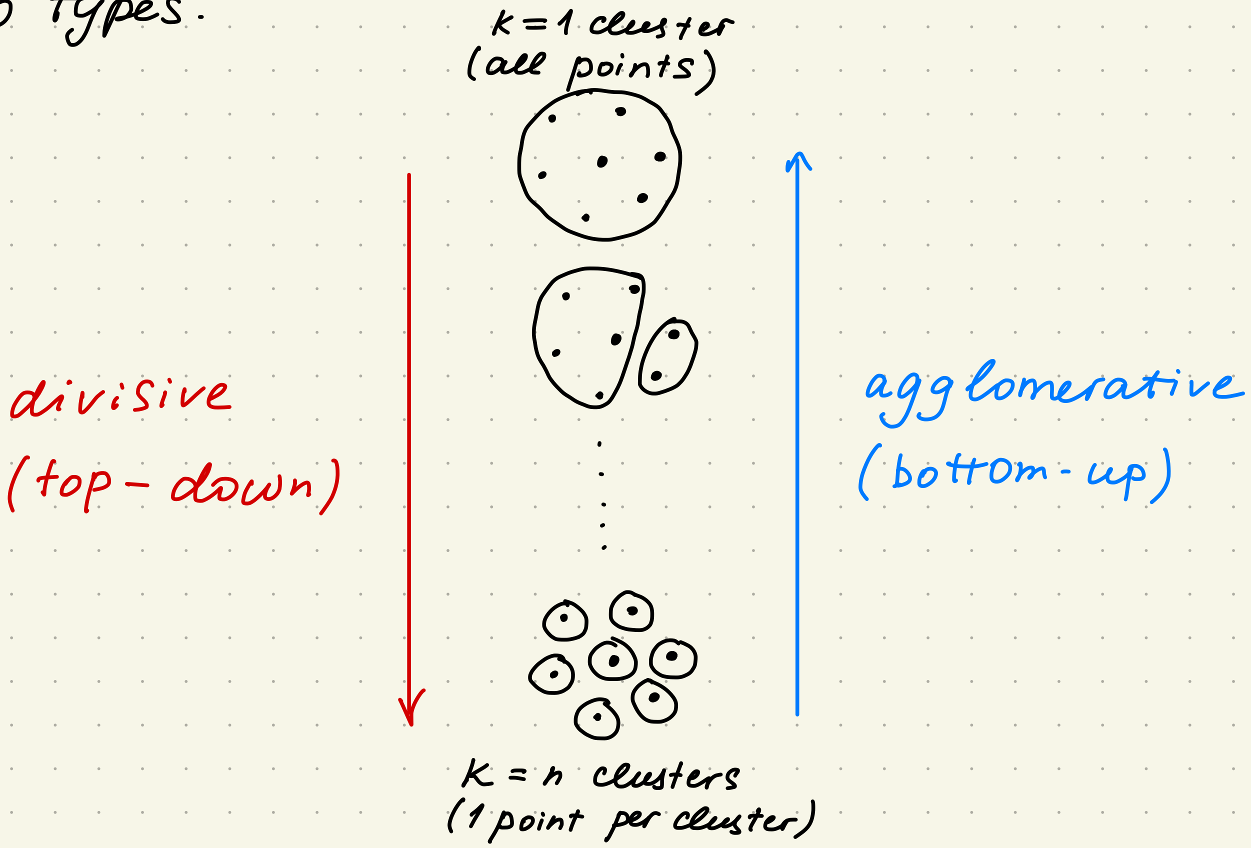
$$Gap(k) \geq Gap(k+1) - S(k+1)$$

adjustment for the variation introduced by the simulation

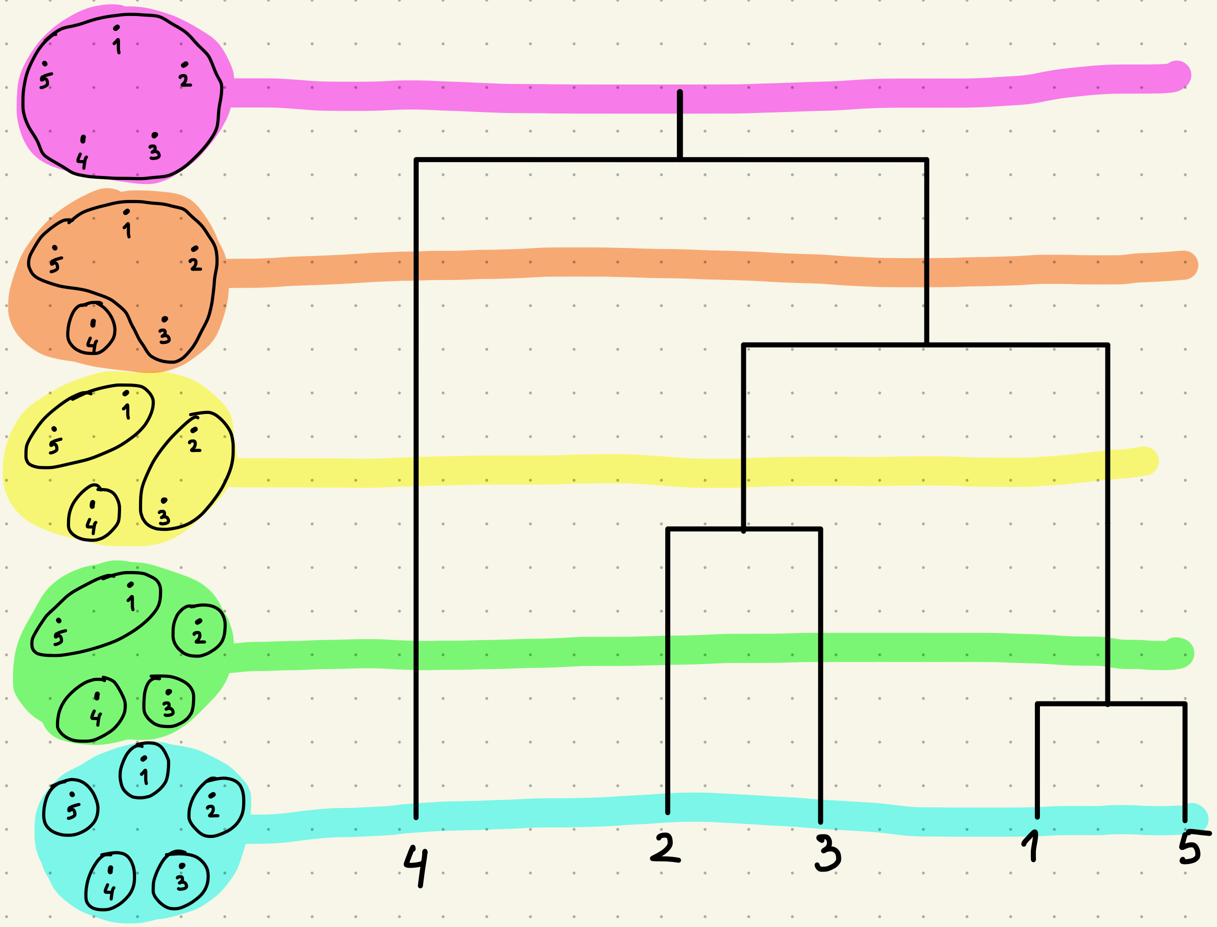


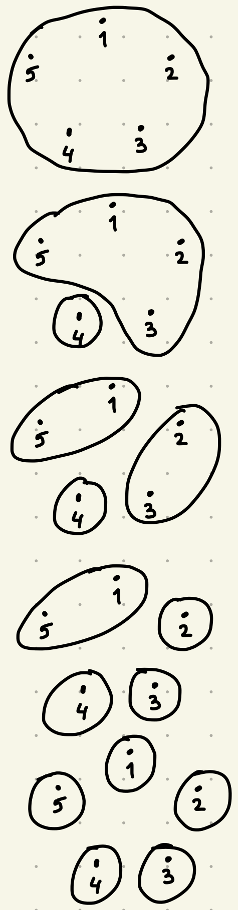
Hierarchical clustering

Two types:

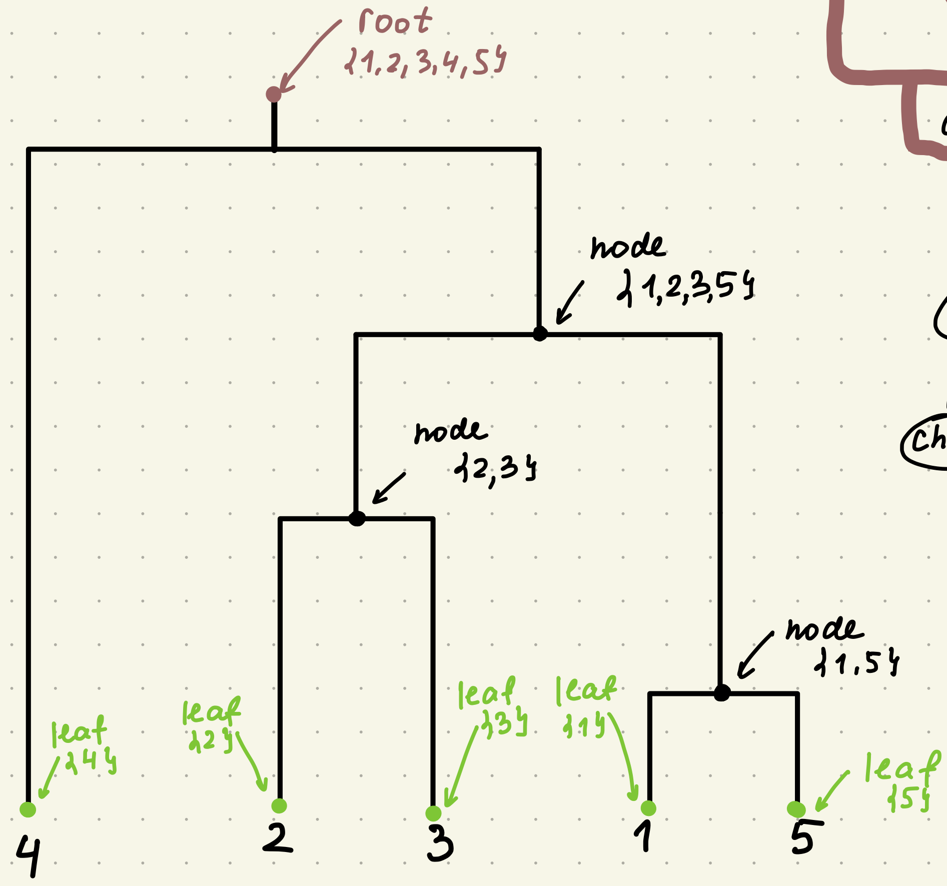


Clustering dendrogram

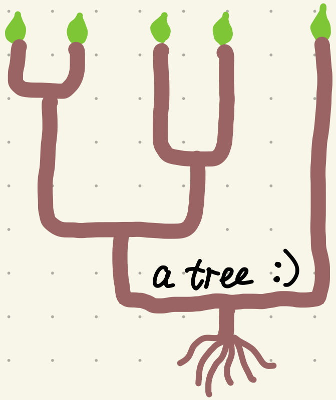




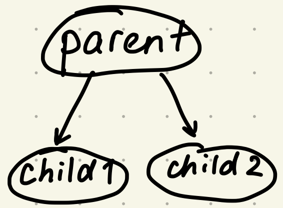
1
2
3
4
5



root
{1,2,3,4,5}
node
{1,2,3,5}
node
{2,3}
node
{1,5}
leaf
{4}
leaf
{2}
leaf
{3}
leaf
{1}
leaf
{5}
4
2
3
1
5



a tree :)



parent
child 1
child 2

Linkages

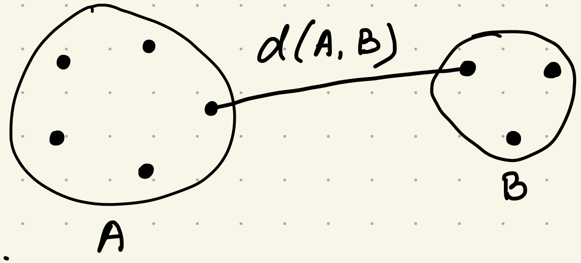
Linkage defines when to merge/split two clusters.

Given $x_1, \dots x_n \in \mathbb{R}^p$ define d_{ij} the dissimilarity between x_i and x_j, e.g. $d_{ij} = \| x_i - x_j \|$.

Given two groups A and $B \subseteq \{1, \dots, n\}$ denote by $n_A = |A|$ and $n_B = |B|$ the size of A and B.

Definde dissimilarity $d(A,B)$ between A and B,

e.g. $d(A,B) = \min\limits_{i \in A, j \in B} d_{ij}$.

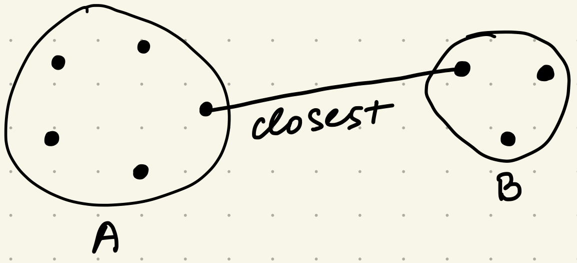


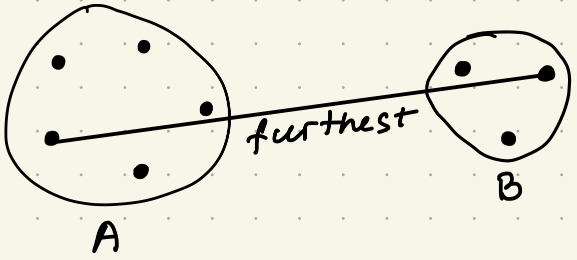
Agglomerative Clustering:

- Start with one point per group.
- Merge A and B with the smallest $d(A,B)$.

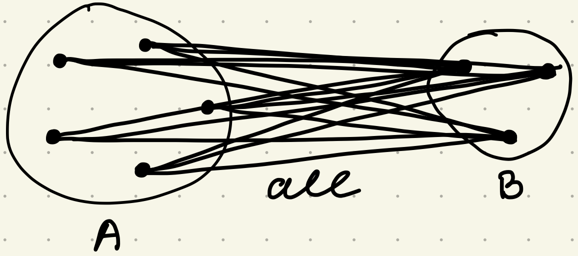
$$d_{single}(A,B) = \min_{i\in A, j\in B} d_{ij}.$$

$$d_{complete}(A,B) = \max_{i\in A, j\in B} d_{ij}.$$

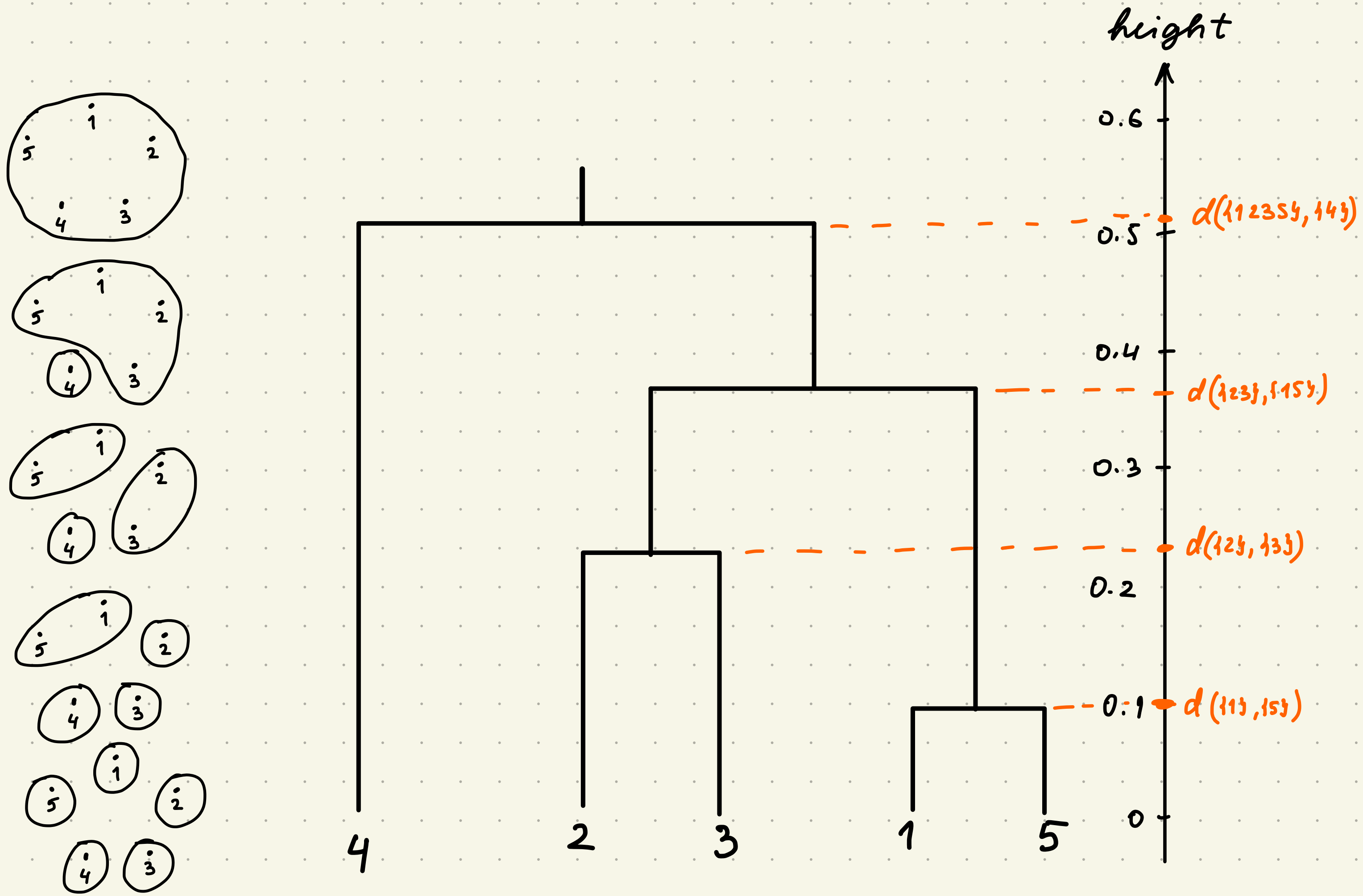




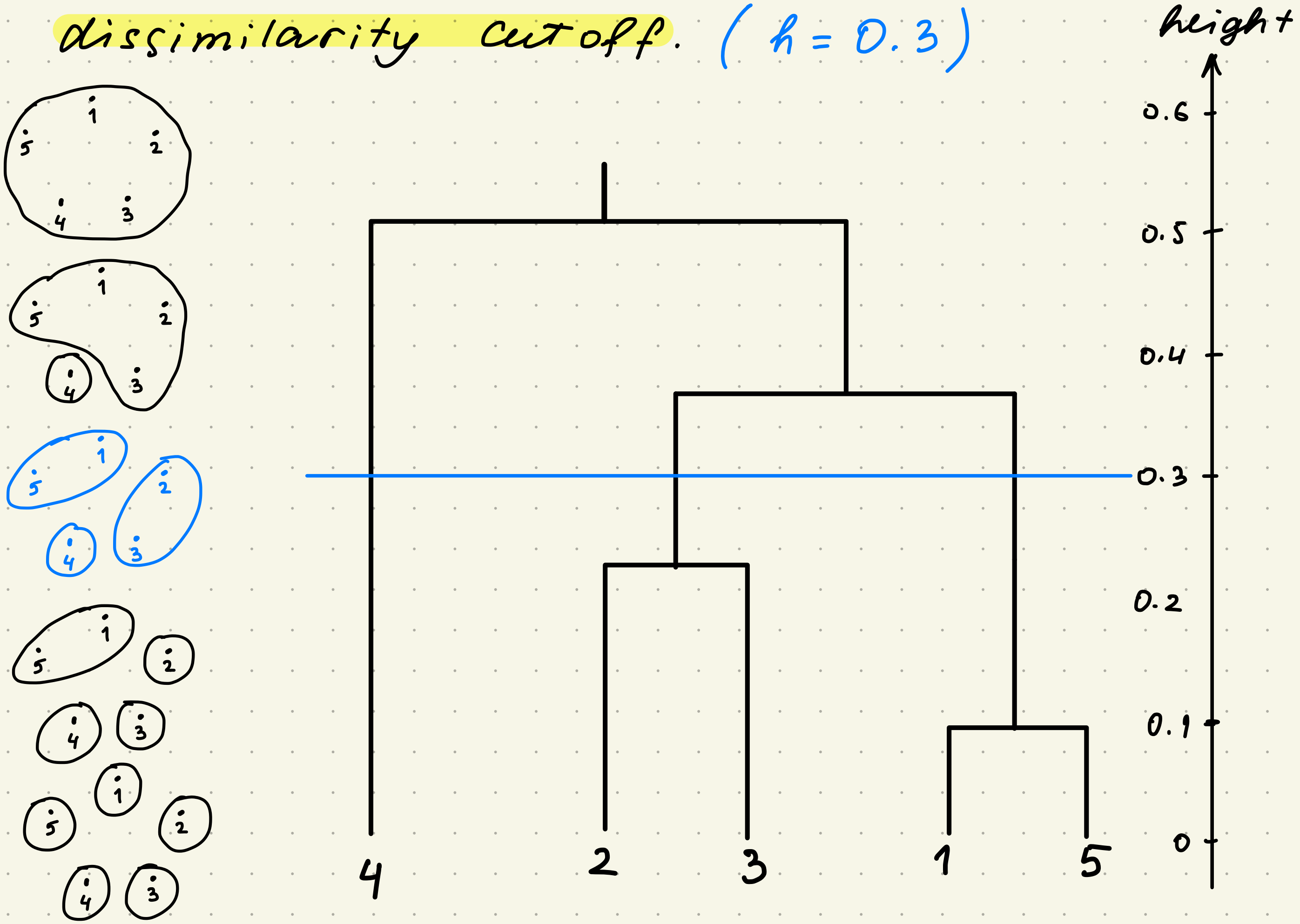
$$d_{average}(A,B) = \frac{1}{n_A n_B} \sum_{i\in A} \sum_{j\in B} d_{ij}.$$



- Single linkage suffers from chaining
- Complete linkage suffers from crowding


height
0.6
0.5
0.4
0.3
0.2
0.1
0
d({1235},{4})
d({23},{15})
d({2},{3})
d({1},{5})
4
2
3
1
5

Given a dendrogram you can decide on the dissimilarity cutoff. ($h = 0.3$)



height
0.6
0.5
0.4
0.3
0.2
0.1
0
4
2
3
1
5