Hierarchical clustering: start

Watch video "Hierarchical Clustering - Fun and Easy Machine Learning" (10min) by Augmented Startups

`https://www.youtube.com/watch?v=EUQY3hL38cw`

(link in MyCourses)

Questions:

- Which linkage metric to use?
- What kinds of clusters can you find?
- Is there anything equivalent to large data?

Generic agglomerative hierarchical clustering

given D = intercluster distance ("linkage metric")

Initialize distance matrix $\mathbf{M}$
Repeat until termination:

1. pick closest pair of clusters C_i and C_j ($D(C_i, C_j)$ minimal)
2. merge clusters: $C_{ij} = C_i \cup C_j$
3. update $\mathbf{M}$
   - remove rows and cols of C_i and C_j
   - add a new row and col for C_{ij} + their entries (distances to C_{ij})

Famous linkage metrics

Single	$\min_{\mathbf{x_1} \in C_1, \mathbf{x_2} \in C_2} \{d(\mathbf{x_1}, \mathbf{x_2})\}$	elongated, straggly, also concentric clusters
Complete	$\max_{\mathbf{x_1} \in C_1, \mathbf{x_2} \in C_2} \{d(\mathbf{x_1}, \mathbf{x_2})\}$	small, compact, hyper-spherical, equal-sized
Average	$\frac{\sum_{\mathbf{x_1} \in C_1, \mathbf{x_2} \in C_2} d(\mathbf{x_1}, \mathbf{x_2})}{\lvert C_1 \rvert \lvert C_2 \rvert}$	quite compact; allows different sizes and densities
Minimum variance (Ward)	$SSE(C_1 \cup C_2) - SSE(C_1)$ $-SSE(C_2)$	compact, quite well-separated, hyperspherical; not elongated or very different sized
Distance of centroids	$d(\mathbf{c_1}, \mathbf{c_2})$	hyperspherical, equal-sized; not elongated

Famous linkage metrics

- linkage metric has a strong effect on results!
- **Warning**: most linkage metrics are sensitive to data order! ⇒ results may change if you shuffle data
- single linkage is not, but it is prone to “chaining effect”

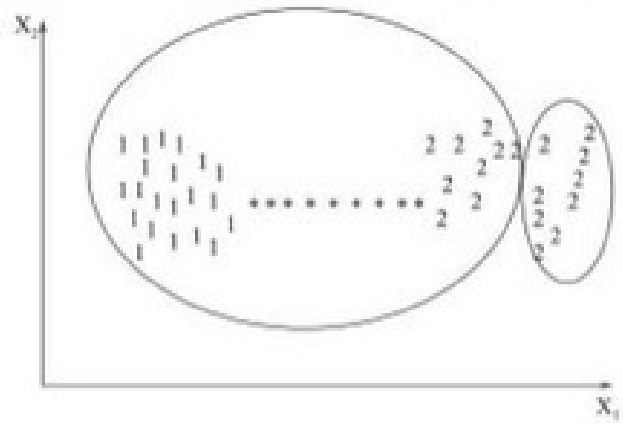


Figure 12. A single-link clustering of a pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*).

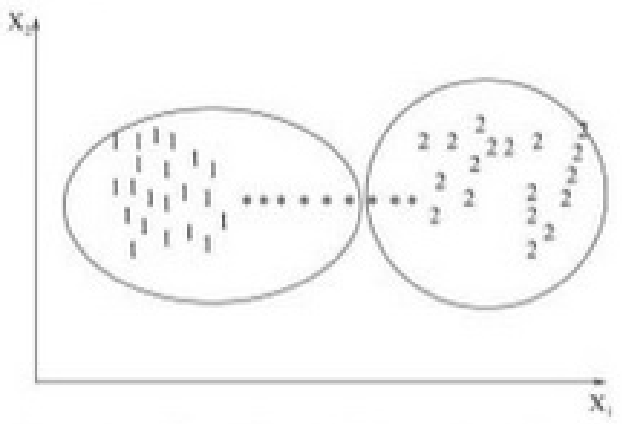


Figure 13. A complete-link clustering of a pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*).

image source https://www.slideshare.net/KalpaGunaratna/incremental-concpetual-clustering-reading-group-discussion

Example (Old Faithful data)

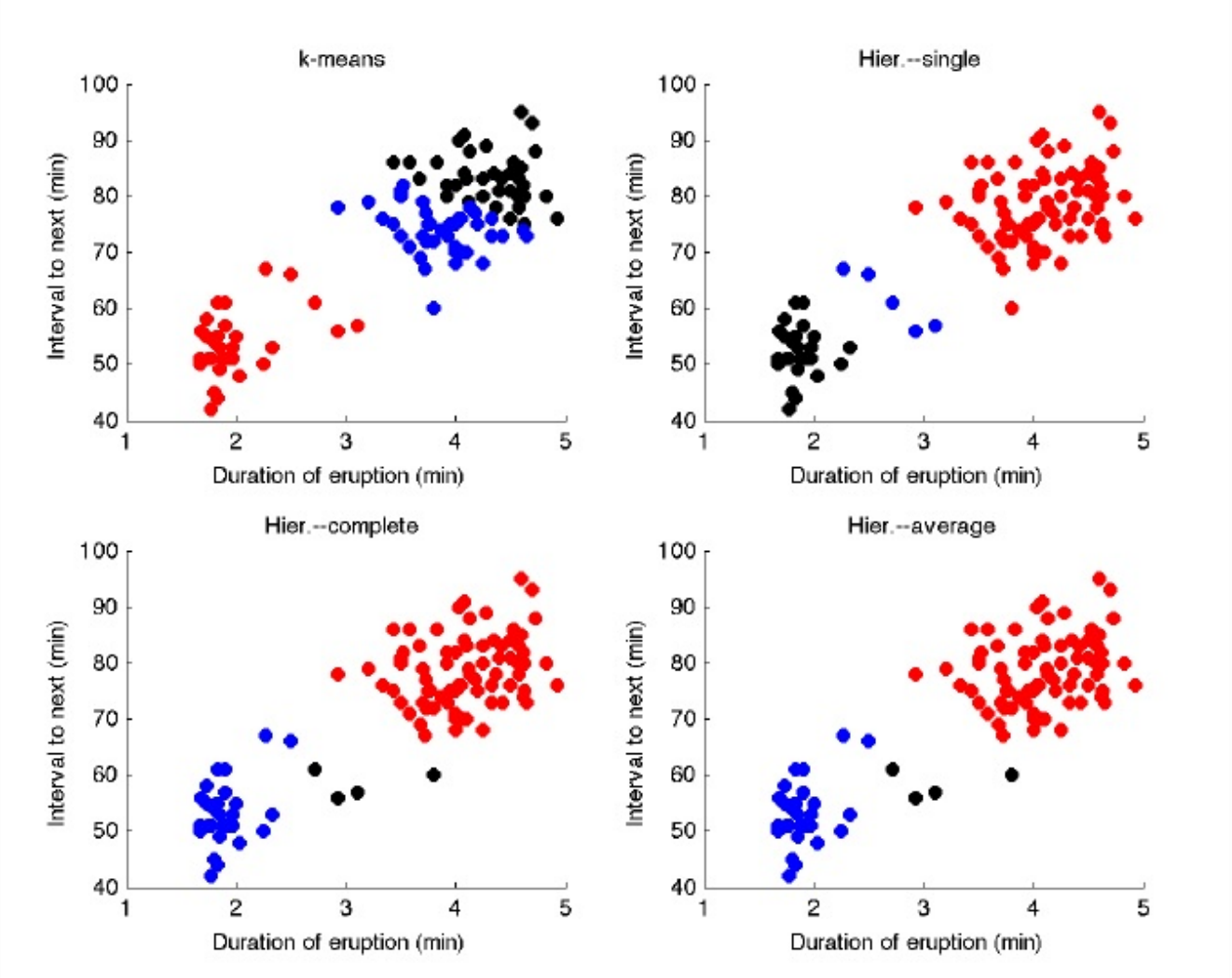


image source: Sungkyu Jung (2013) slides on clustering (STAT2221, Univ. of Pittsburgh)
https://www.stat.pitt.edu/sungkyu/course/2221Fall13/lec7_clustering.pdf

Connection to graph theory

Single linkage is related to **connected components** and complete linkage to **cliques**

Let $e_1, \ldots, e_m$ be edges of complete distance graph with weights $d_1 < d_2 < \ldots < d_m$. ($m = \frac{n(n-1)}{2}$, n data points)

Single linkage

1. Initialize: Create graph $\mathbf{G}$ without edges
   - i.e., n connected components and all data points in their own clusters
2. Repeat until one connected component
   - add new edge e_i with smallest d_i to $\mathbf{G}$
   - form clusters from connected components of $\mathbf{G}$

Connection to graph theory

Complete linkage

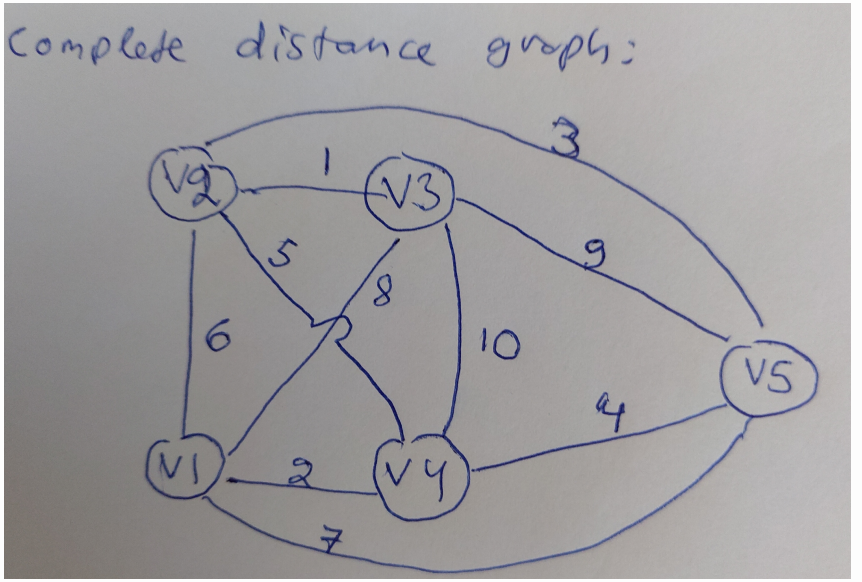
1. Initialize: Create graph $\mathbf{G}$ without edges
   - all data points in their own clusters
2. Repeat until $\mathbf{G}$ complete
   - add new edge e_i with smallest d_i to $\mathbf{G}$
   - if two of the current clusters form a clique in $\mathbf{G}$, merge them

Note: You are not allowed to break existing clusters, even if you would find alternative cliques

Task: graph-based single linkage clustering

Distance matrix:

	v1	v2	v3	v4	v5
v1	0	6	8	2	7
v2	6	0	1	5	3
v3	8	1	0	10	9
v4	2	5	10	0	4
v5	7	3	9	4	0



Add edges in the increasing order of weights $(2, 3, \ldots, 7)$. What are the corresponding single linkage clusters? Complete linkage clusters? (hometask)

Task

*Extra: single linkage clustering from minimum spanning trees

Begin from complete distance graph $\mathbf{G}$ and search its minimum spanning tree (MST)

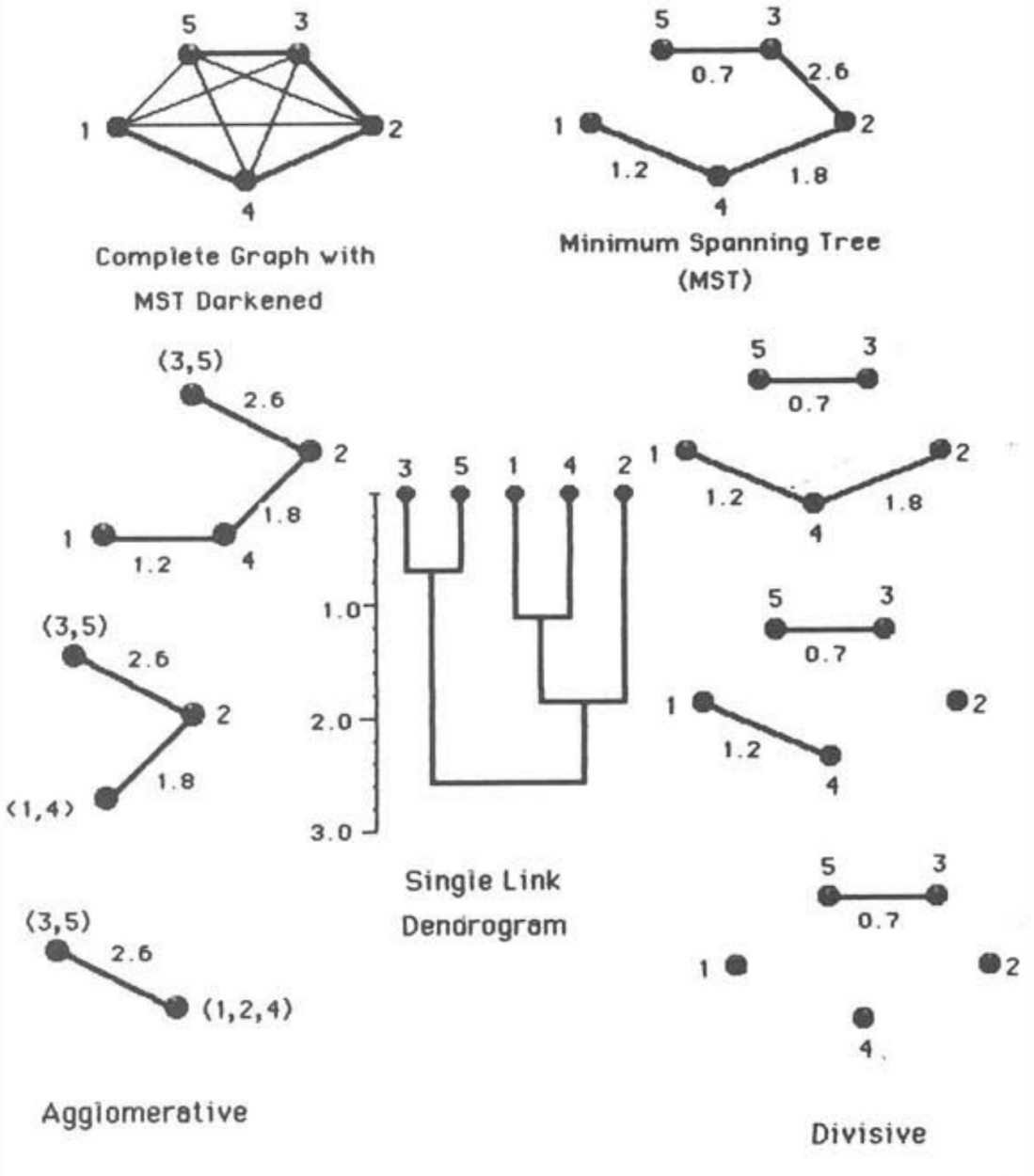
Repeat until all objects belong to one cluster:

1. Merge two clusters that are connected in the MST and have the smallest edge weight
2. Set the edge weight as ∞

Notes:

- The same can be done in a divisive manner: cut the MST edges in the descending order by weight.
- If there are no proximity ties, the result is the same as normal single likage clustering

*Example (Jain and Dubes 1988, Fig 3.6)



Agglomerative or divisive?

- Agglomerative = bottom-up
  - cheaper and easier to implement
  - still slow, at least $O(n^2)$
  - early decisions based on local patterns, cannot cancel later
- Divisive = top-down
  - often better quality clustering
    $\Leftarrow$ large clusters created early, based on global distribution
  - fastest $O(n^2 \log(n))$

Bisecting K-means

Idea: combine divisive hierarchical and K-means.
Given K and q=number of iterations

1. Initialization: put all data points into one cluster
2. Repeat until K clusters:
   - choose cluster C to split (with largest SSE)
   - split C q times with 2-means
   - keep the best split (two new clusters)

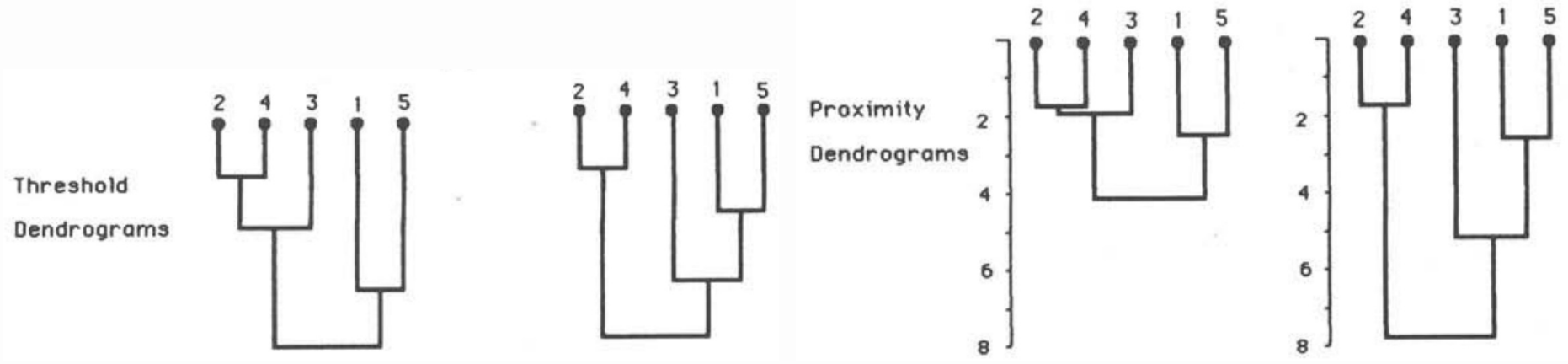
\+ efficient (like K-means)

\+ good results (comparable to hierarchical)

On dendrograms

- **Threshold dendrograms**: in which order the clusters were formed
- **Proximity dendograms**: at which proximities they were formed

	x_2	x_3	x_4	x_5
x_1	5.8	4.2	6.9	2.6
x_2		6.7	1.7	7.2
x_3			1.9	5.6
x_4				7.6



single complete single complete

image source Jain & Dubes 1988 Fig 3.5

Dendrogram example

Here real (biological) classes of data points are shown under the dendrogram

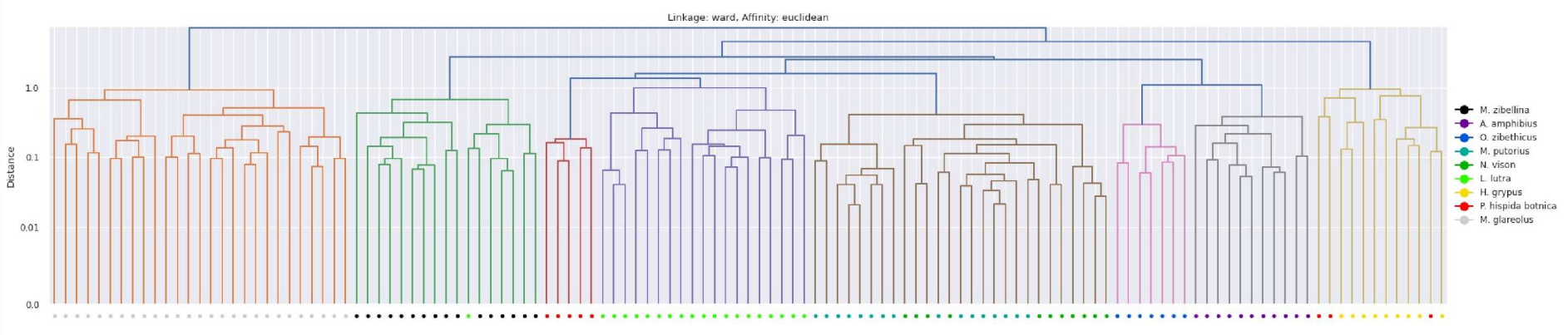


image author Lehtiniemi 2020

Another dendrogram example

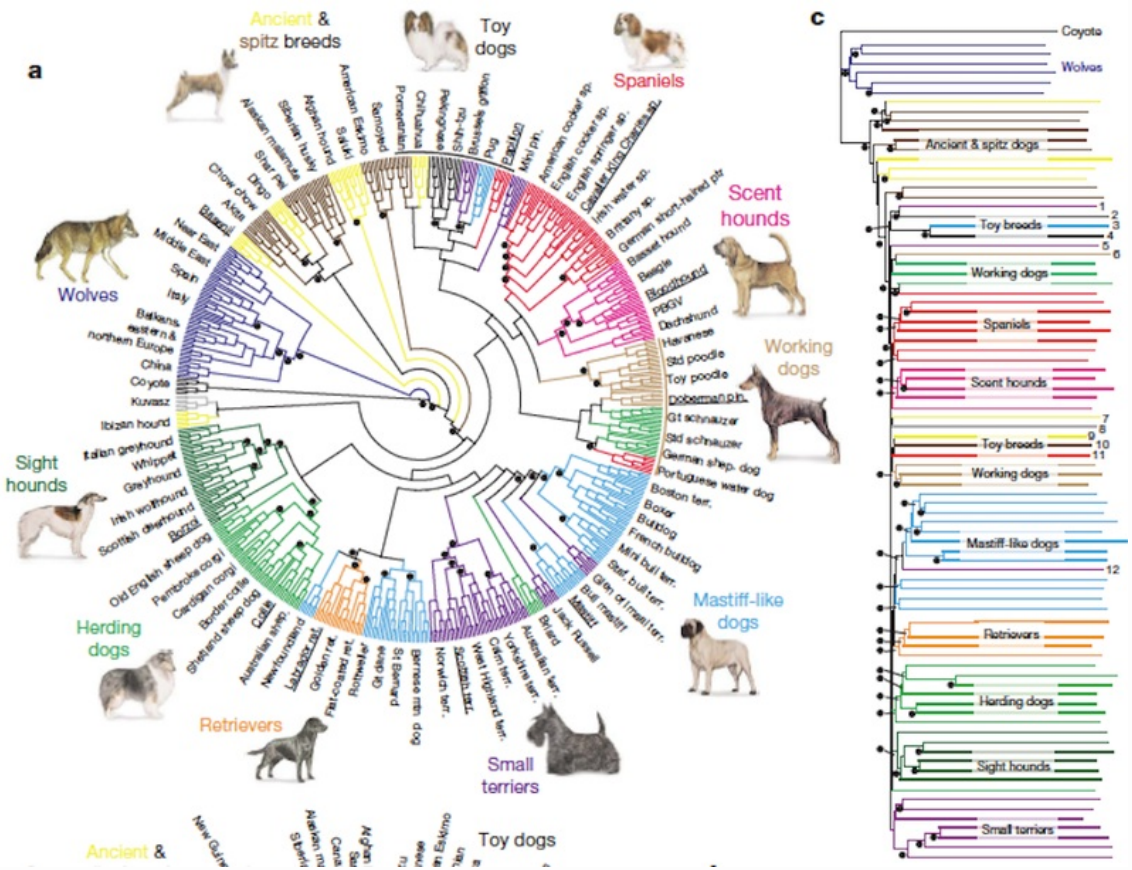


image source
https://www.instituteofcanineblology.org/how-to-read-a-dendrogram.html

Summary

- useful information on clustering structure
  - dendrograms!
- linkage metrics have a strong effect
  - Beware: most metrics sensitive to data order!
- connections to graph theory (single $\leftrightarrow$ connected components, complete $\leftrightarrow$ cliques)
- inefficient for really large data (at least $O(n^2)$ time and space)

Voluntary task: Fill a summary table!

method	data type	cluster type	benefits	drawbacks
K-representatives				
K-means				
K-medoids				
...				
Hierarchical				
singe-link				
...				
Graph-based				
Density-based				
Probabilistic				

Further reading

- Gan, Ma, Wu: Data clustering – theory, algorithms, and applications. SIAM 2007.
- Jain and Dubes: Algorithms for clustering data. Prentice-Hall 1988.

Today's lecture

1. K-representatives clustering
   - Recap K-means (video)
   - other members of the family
2. Hierachical clustering
   - introduction (video)
   - more on linkage metrics, connections to graph theory, dendrograms

Book 6.3, 6.4

Main groups of clustering methods (Aggarwal)

- Representative-based
- Hierarchical
- Probabilistic model-based
- Density-based (including grid-based)
- Graph-based
- Matrix factorization based

Representative-based: K-means

Watch video "K-means clustering: how it works" (7.5 min) by Victor Lavrenko

https://www.youtube.com/watch?v=_aWzGGNrcic

Questions

- Why K-means is only for numerical data?
- Could we apply something similar to categorical data? or other data types?

K-means

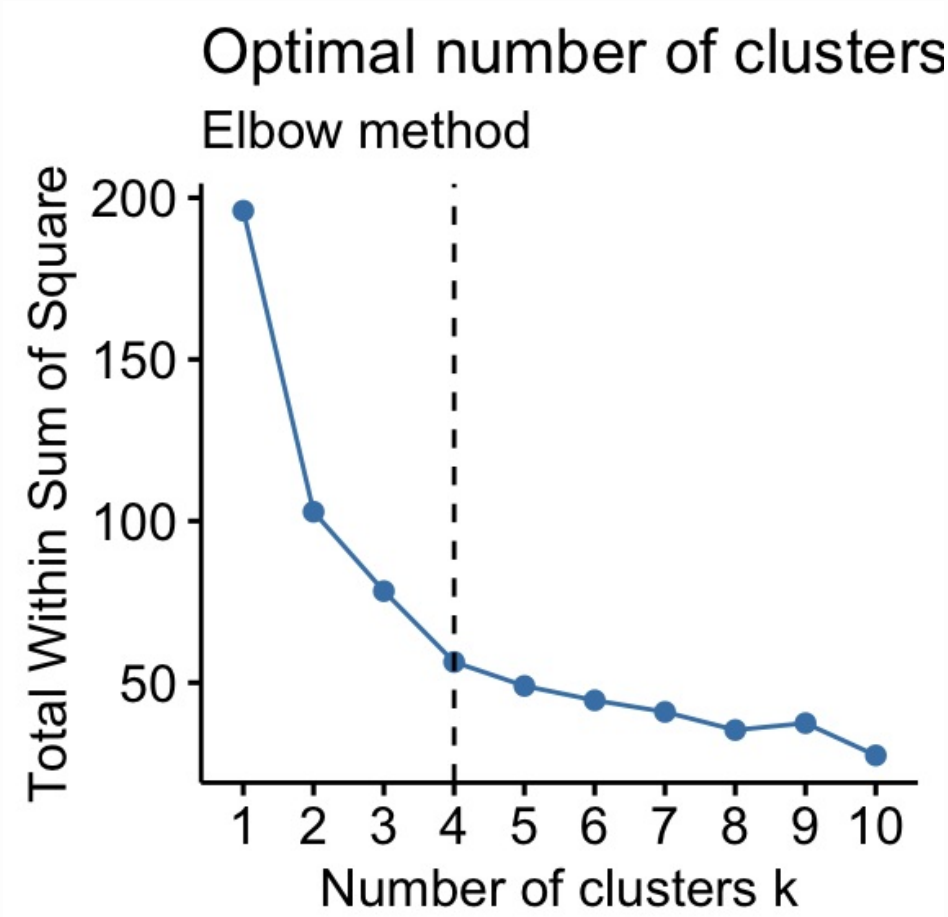
Notations: Data points $\mathbf{x}_i \in \mathcal{D}$, clusters $C_1, \ldots, C_K$, centroids $\mathbf{c}_1, \ldots, \mathbf{c}_k$, $\mathbf{m}$ mean of data.

- objective: minimize $SSE = \sum_{j=1}^{K} \sum_{\mathbf{x} \in C_j} L_2^2(\mathbf{x}, \mathbf{c}_j)$
  - minimizes wc, maximizes bc, since $\sum_{\mathbf{x} \in \mathcal{D}} L_2^2(\mathbf{x}, \mathbf{m}) = \sum_{j=1}^{K} \sum_{\mathbf{x} \in C_j} L_2^2(\mathbf{x}, \mathbf{c}_j) + \sum_{j=1}^{K} |C_j| L_2^2(\mathbf{c}_j, \mathbf{m})$
- tends to find **compact, hyperspherical** clusters
- **designed only for** L_2, but many K-representative variants for other distance measures
  - **warning**: if you use another distance in K-means, may not find even local optimum or converge. **Why?**
- very sensitive to the initialization of centroids! $\rightarrow$ **run multiple times**

K-means

- \+ can produce good results if clusters compact, well-separated, hyperspherical
- \+ easy to implement
- \+ quite efficient $O(nKq)$, q=number of iterations
- – basic form requires L_2 measure
- – sensitive to outliers
- – sensitive to initialization (some improved strategies)
- – converges to local optimum (not necessarily global)
- – sometimes convergation can be slow
- – needs parameter K

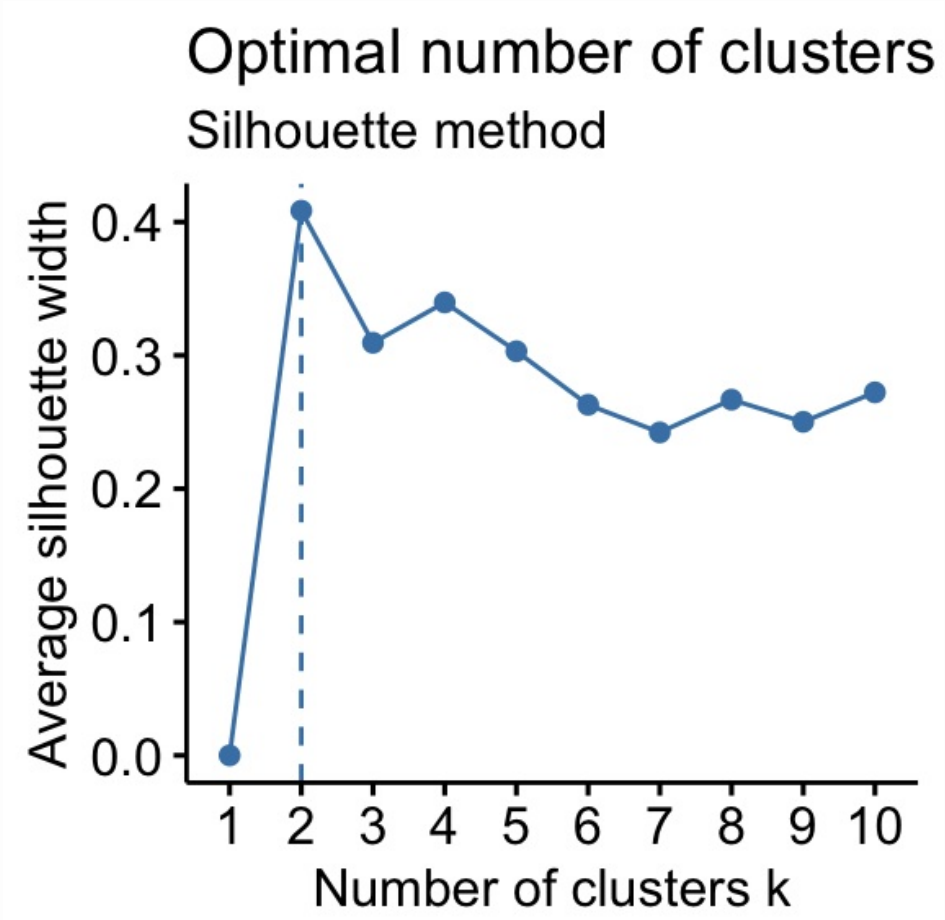
Choosing number of clusters: SSE *elbow*



- SSE decreases with K
- is there an elbow of the curve, where speed slows down?
- not always clear

source https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/

Choosing number of clusters: silhouette peak



- Silhouette tells how well an individual data point is clustered
- **Average silhouette** evaluates the entire clustering

source https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/

Silhouette coefficient

Silhouette of a point $\mathbf{x}$ is

$$S(\mathbf{x}) = \begin{cases} 0 & \text{if singleton} \\ \frac{b-a}{\max\{a,b\}} & \text{otherwise} \end{cases}$$

a=mean distance of $\mathbf{x}$ to points in the same cluster

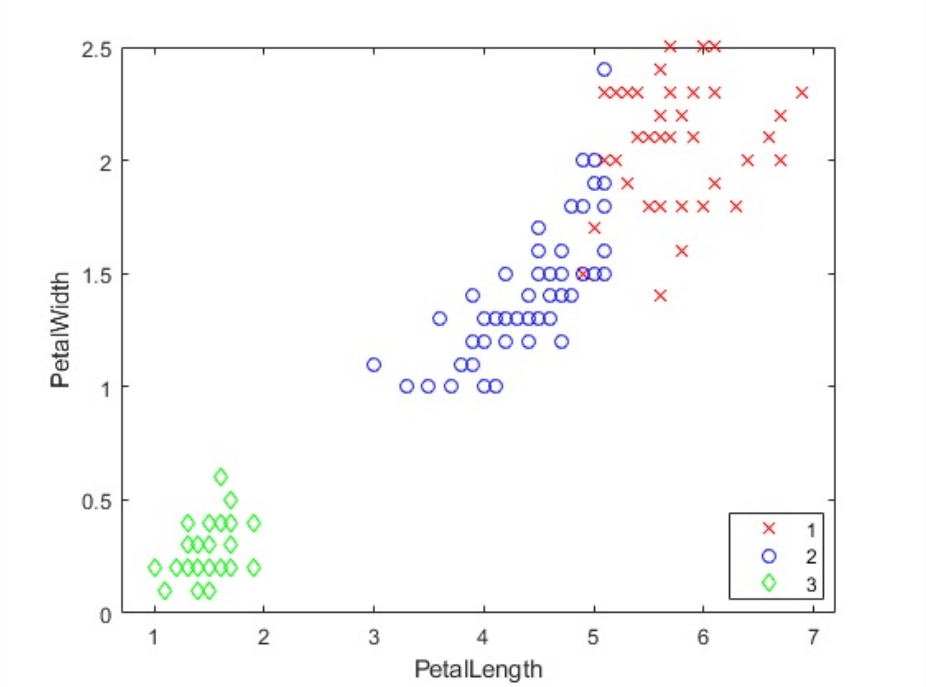
b=mean distance of $\mathbf{x}$ to points in the closest neighbouring cluster

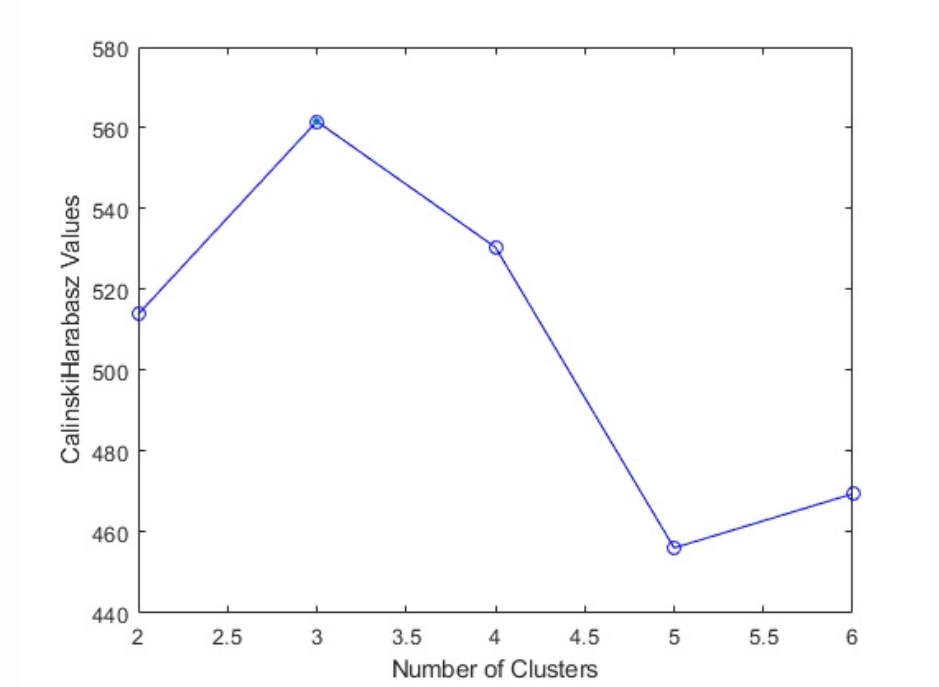
$\Rightarrow$ average Silhouette $S_{avg} = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{D}} S(\mathbf{x})$

$\rightarrow$ More on lecture 5

Choosing number of clusters: Calinski-Harabasz

based on inter-cluster and intra-cluster variances





source https://www.mathworks.com/help/stats/clustering.evaluation.calinskiharabaszevaluation-class.html

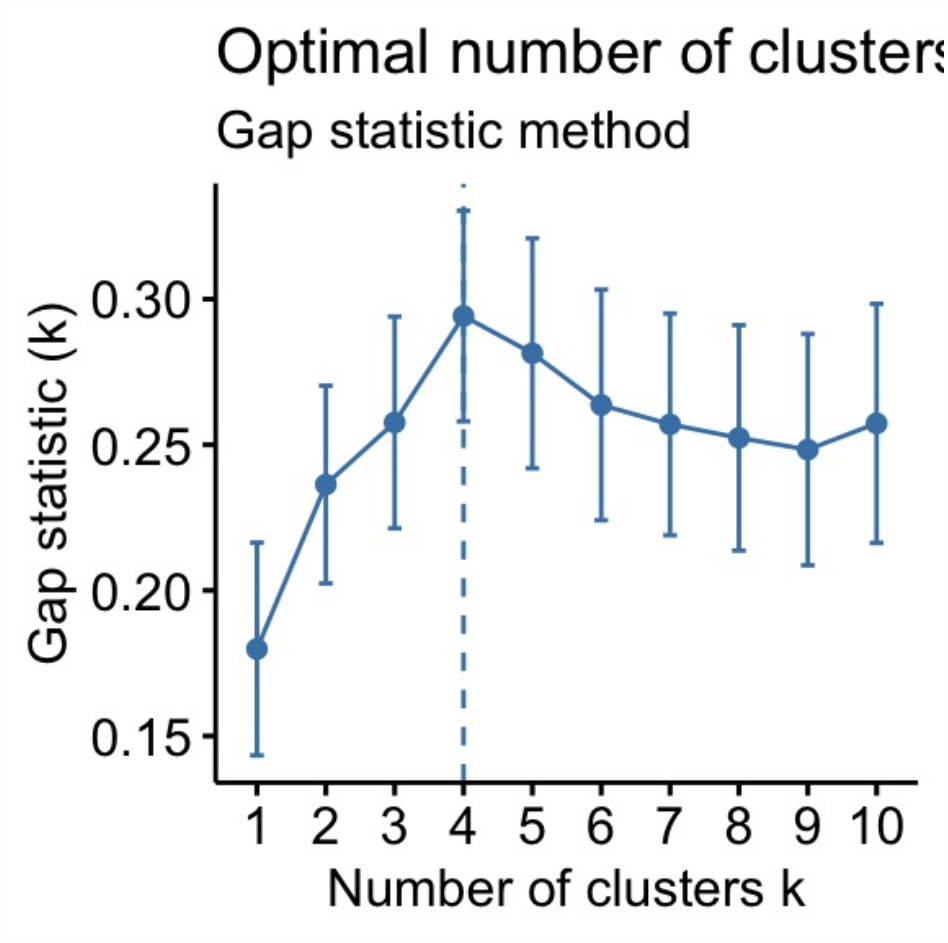
Choosing number of clusters: Calinski-Harabasz

$$S_{CH} = \frac{(n-K)B}{(K-1)W}$$

- between-cluster variance $B = \sum_{i=1}^{K} |C_i| L_2^2(\mathbf{c}_i, \mathbf{m})$ ($\mathbf{m}$ = mean of the whole data)
- within-cluster variance $W = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} L_2^2(\mathbf{x}, \mathbf{c}_i)$
- well suitable to K-means!

→ More on lecture 5

Choosing number of clusters: Gap statistic



- Cluster data and evaluate $W_K = \sum_{r=1}^{K} \frac{1}{2|C_r|} \sum_{\mathbf{x},\mathbf{y} \in C_r} d(\mathbf{x}, \mathbf{y})$
- Evaluate W_K in B random data sets $\rightarrow$ $W_{K1}, \ldots, W_{KB}$
- $Gap(K) = \frac{1}{B} \sum_{b=1}^{B} \log(W_{Kb}) - \log(W_K)$
- Choose $\min K$: $Gap(K) \geq Gap(K+1) - \sigma_{K+1}$

source https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/

Gap statistic

- σ_K= standard deviation of $W_{K1}, \ldots, W_{KB}$
- if $d = L_2^2$, W_K estimates SSE

\+ suits to **any clustering method and distance** d

– computationally heavy (B random simulations for all tested K)

Further reading: Tibshirani et al.: Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society, 2001.

K-means extensions

- **K-medians**
  - uses L_1 measure and medians
  - determine median values along each dimension separately
  - \+ more robust to outliers
  - – computationally more costly
- **K-medoids**
  - medoid = the center-most **data point** in a cluster
  - \+ more efficient (but slower than k-means)
  - \+ allows any distance function
  - \+ suits to any data type! (given distance function)

K-means vs. K-medoids

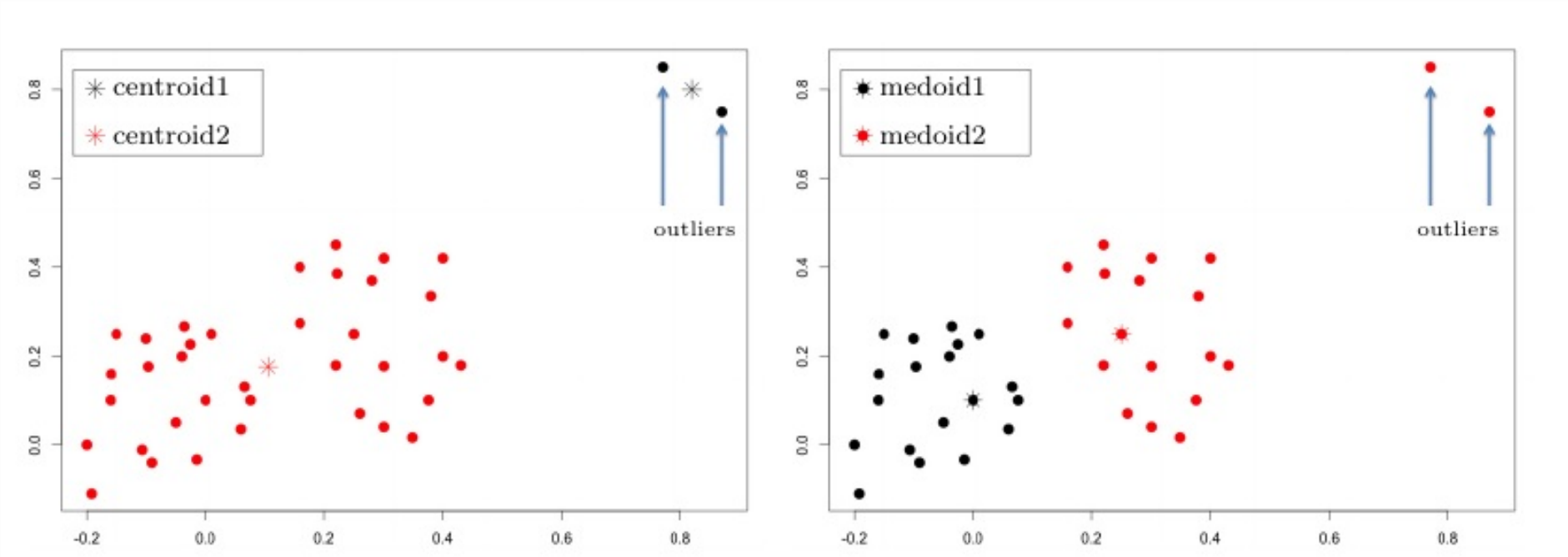


Figure 3.3: Outliers effect: k-means clustering (left) vs. k-medoids clustering (right)

image source: Soheily-Khah (2016): Generalized k-means based clustering for temporal data under time warp

K-modes

- for categorical data
- minimize $\sum_{\mathbf{x} \in C} \sum_{i=1}^{k} d_s(x_i, c_i)$, where

$$d_s(x_i, y_i) = \begin{cases} 1 & \text{if } x_i \neq y_i \\ 0 & \text{otherwise} \end{cases}$$

- "simple matching distance" = overlap distance without weights
- cluster centers $\mathbf{c}$ are "modes" (choose most frequent values of each feature)

K-modes: example

K=3. Original centers ("modes") individuals 1, 5, 10

Individual	Q1	Q2	Q3	Q4	Q5	C1	C2	C3
1	A	B	A	B	C	0	4	2
2	A	A	A	B	B	2	4	4
3	C	A	B	B	A	4	2	4
4	A	B	B	A	C	2	5	0
5	C	C	C	B	A	4	0	5
6	A	A	A	A	B	3	5	4
7	A	C	A	C	C	2	4	3
8	C	A	B	B	C	3	3	3
9	A	A	B	C	A	4	4	3
10	A	B	B	A	C	2	5	0

Note: Many ways to choose initial “modes”.

K-modes: example

Calculate new modes:

Cluster	Q1	Q2	Q3	Q4	Q5
1 (1), (2), (6), (7), (8)	A	A	A	B	C
2 (3), (5)	C	A	B	B	A
3 (4), (9), (10)	A	B	B	A	C

Example from "K-Modes intuition and example" by Aysan Fernandes
https://www.youtube.com/watch?v=b39_vipRkUo

K-prototypes

- for mixed data
- minimize

  $\sum_{\mathbf{x}\in C}\left(\sum_{i=1}^{q}(x_i - c_i)^2 + \gamma \sum_{i=q+1}^{k} d_s(x_i, c_i)\right)$, where

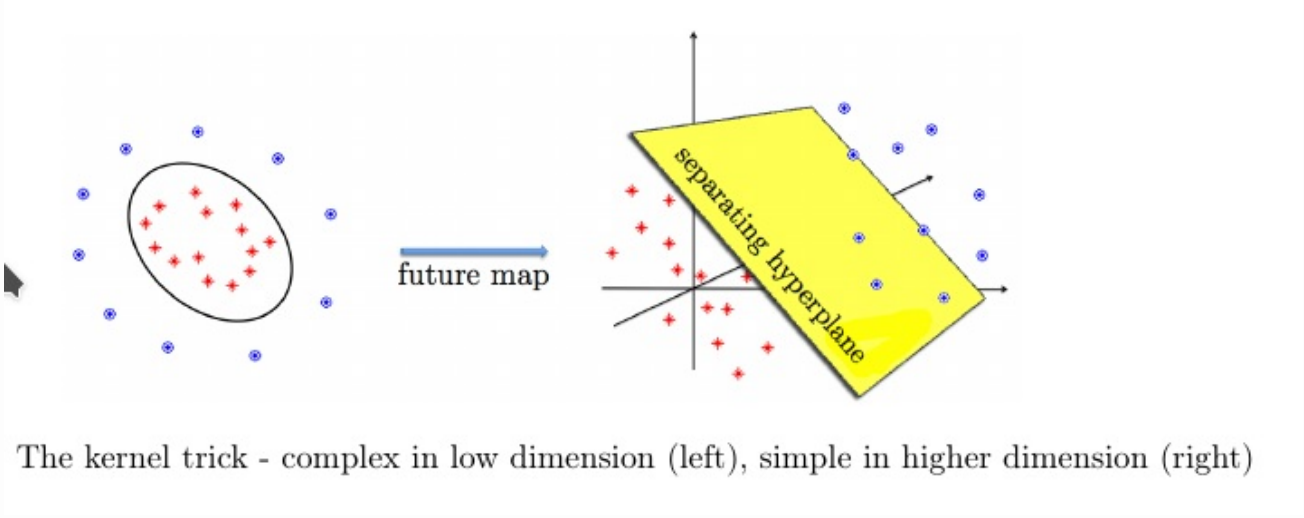
  $x_1 \ldots, x_q$ numerical values

  $x_{q+1}, \ldots, x_k$ categorical values

  γ=balancing weight
- cluster centroids $\mathbf{c}$ are "prototypes"

K-means extensions: Kernel-K-means

Idea: map data implicitely to a higher dimensional space and perform K-means there



The kernel trick - complex in low dimension (left), simple in higher dimension (right)

\+ robust

\+ can detect arbitrary shapes

\– expensive

image source Soheily-Khah (2016): Generalized k-means based clustering for temporal data under time warp

Summary

- Basic idea of K-representatives method
  - K-means, K-medians, K-medoids, K-modes, K-prototypes
- Techniques to choose K
  - SSE elbow, Silhouette peak, Calinski-Harabasz, Gap statistic

Further reading:

- Gan, Ma, Wu: Data clustering – theory, algorithms, and applications. SIAM 2007.
- Jain and Dubes: Algorithms for clustering data. Prentice-Hall 1988.