STAT 215A Fall 2022 Week 5

Theo Saarinen

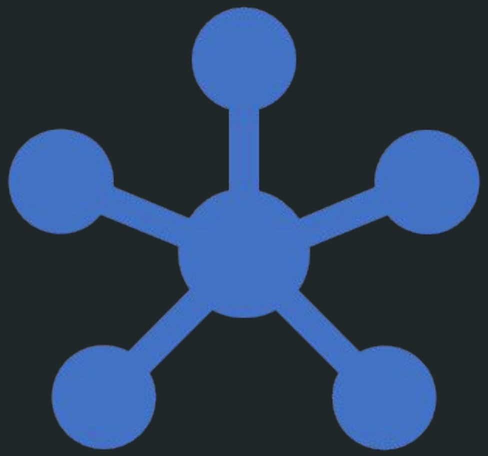
Announcements

- Lab 1 due **tonight at 11:59pm**
- Lab 2 + Homework 2 will be released today
  - Due in two weeks: **October 7 11:59pm**
- Peer reviews will be sent out this coming Monday (**Sep 26**)
  - Due in 1 week: **October 3 11:59pm**
  - I will push 2 other reports to your Github repo. Will include instructions + template.

Plan for today:

- K-means
- Hierarchical Clustering
- NMF
- Centering and Scaling?
- Introduce Lab 2

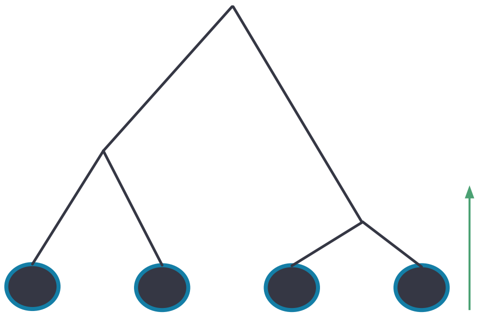
Clustering



Hierarchical clustering

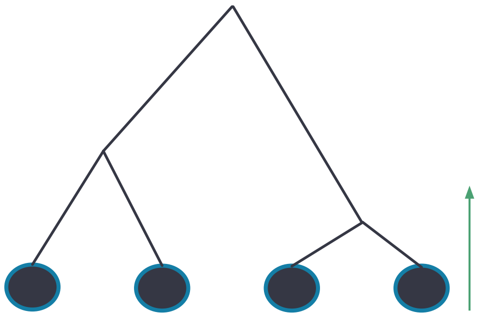
Hierarchical clustering

- Gives family of nested clusterings, presented as a tree



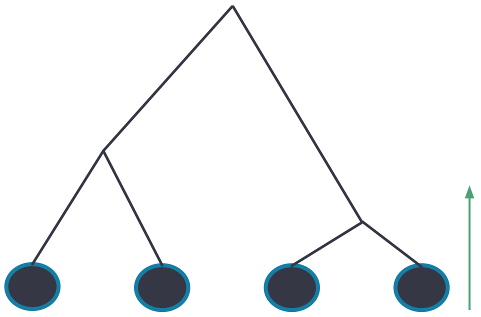
Hierarchical clustering

- Gives family of nested clusterings, presented as a tree
- A **greedy**, agglomerative algorithm, not based upon an optimization problem



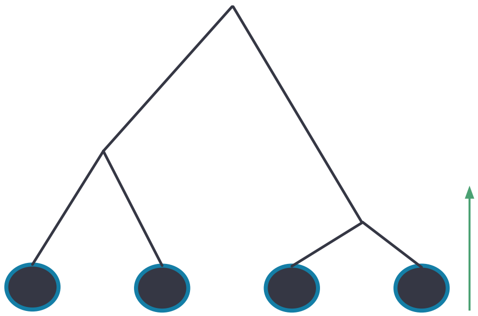
Hierarchical clustering

- Gives family of nested clusterings, presented as a tree
- A **greedy**, agglomerative algorithm, not based upon an optimization problem
- At the **lowest level**, each cluster contains a **single observation**



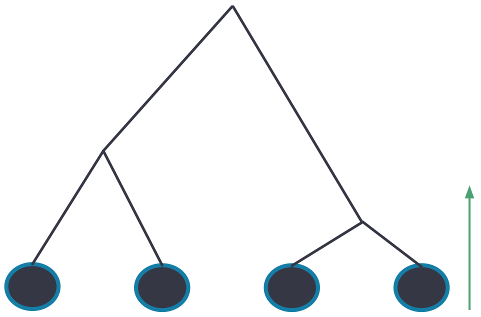
Hierarchical clustering

- Gives family of nested clusterings, presented as a tree
- A **greedy**, agglomerative algorithm, not based upon an optimization problem
- At the **lowest level**, each cluster contains a **single observation**
- As we move up the tree, some leaves begin to fuse into branches – these are observations that are similar to each other.
    - The lower in the tree the fusion occurs, the more similar the groups of observations are to each other

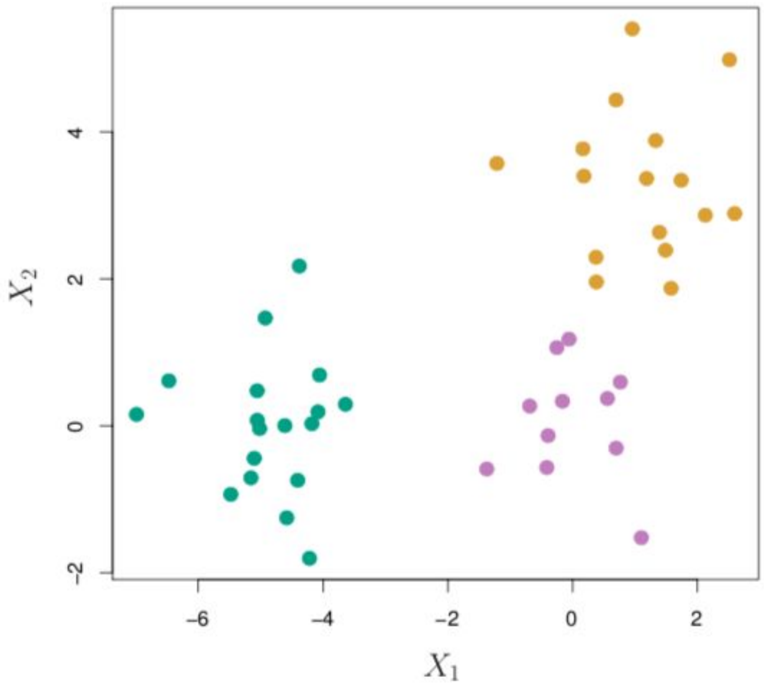


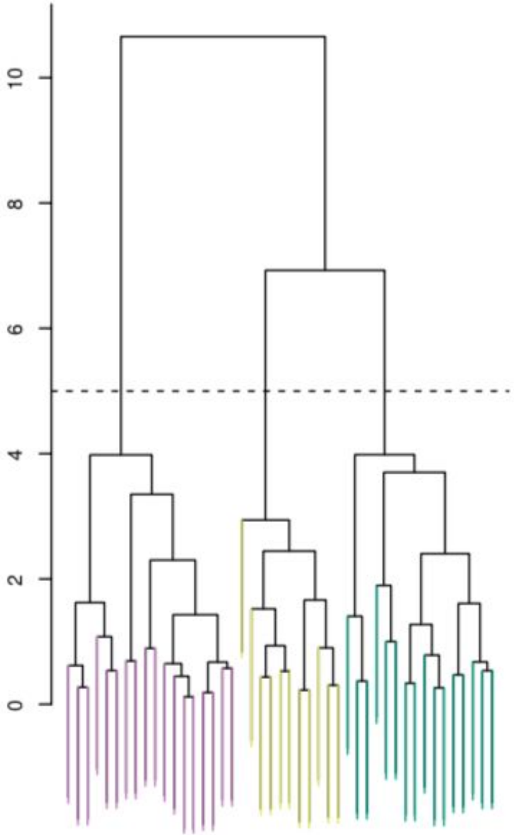
Hierarchical clustering

- Gives family of nested clusterings, presented as a tree
- A **greedy**, agglomerative algorithm, not based upon an optimization problem
- At the **lowest level**, each cluster contains a **single observation**
- As we move up the tree, some leaves begin to fuse into branches – these are observations that are similar to each other.
  - The lower in the tree the fusion occurs, the more similar the groups of observations are to each other
- At the **highest level**, there is **only one cluster** containing all observations



Interpreting dendrograms





How to join clusters/observations

1. **Distance metric:** a measure of dissimilarity between two observations

$$d(x, y)$$

   - Examples: L^2, L^1, your favorite norm, $1 - cor(x, y)$

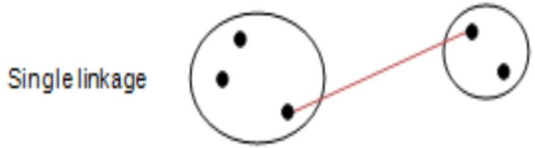
How to join clusters/observations

1. **Distance metric:** a measure of dissimilarity between two observations

$$d(x, y)$$

   - Examples: L^2, L^1, your favorite norm, $1 - cor(x, y)$

2. **Linkage metric:** rule for joining two clusters
   - Single linkage



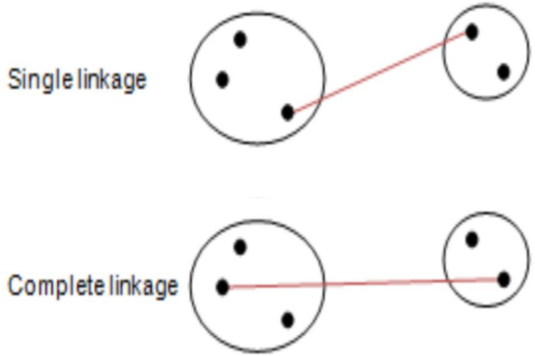
How to join clusters/observations

1. **Distance metric:** a measure of dissimilarity between two observations

$$d(x, y)$$

   - Examples: L^2, L^1, your favorite norm, $1 - cor(x, y)$

2. **Linkage metric:** rule for joining two clusters
   - Single linkage
   - Complete linkage



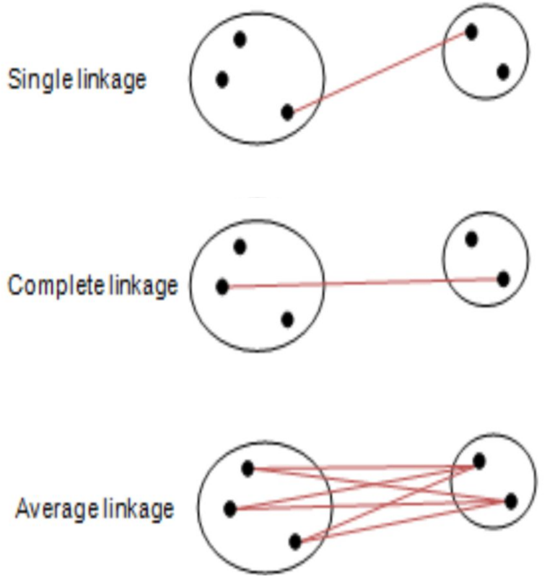
How to join clusters/observations

1. **Distance metric:** a measure of dissimilarity between two observations

$$d(x, y)$$

   - Examples: L^2, L^1, your favorite norm, $1 - cor(x, y)$

2. **Linkage metric:** rule for joining two clusters
   - Single linkage
   - Complete linkage
   - Average linkage



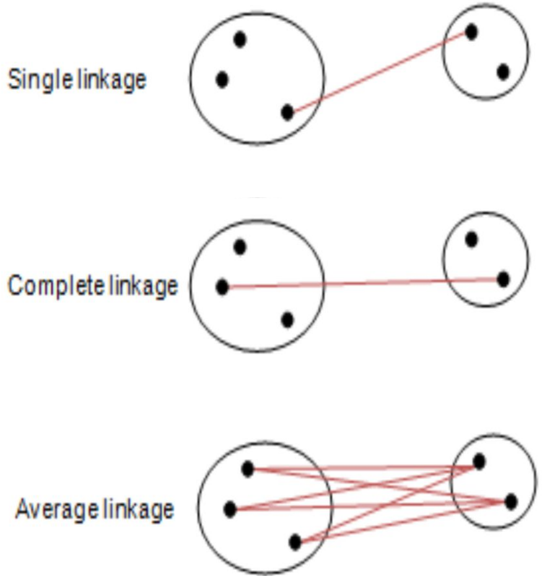
How to join clusters/observations

1. **Distance metric:** a measure of dissimilarity between two observations

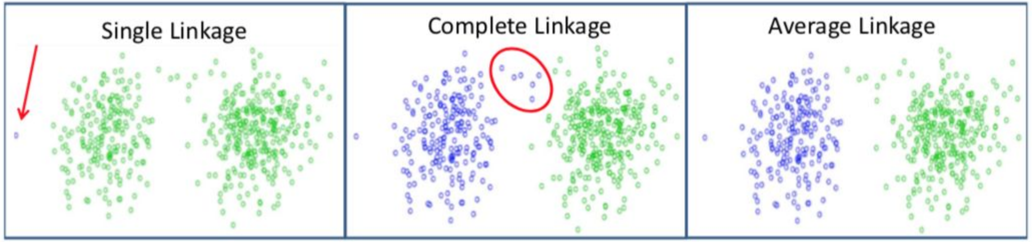
$$d(x, y)$$

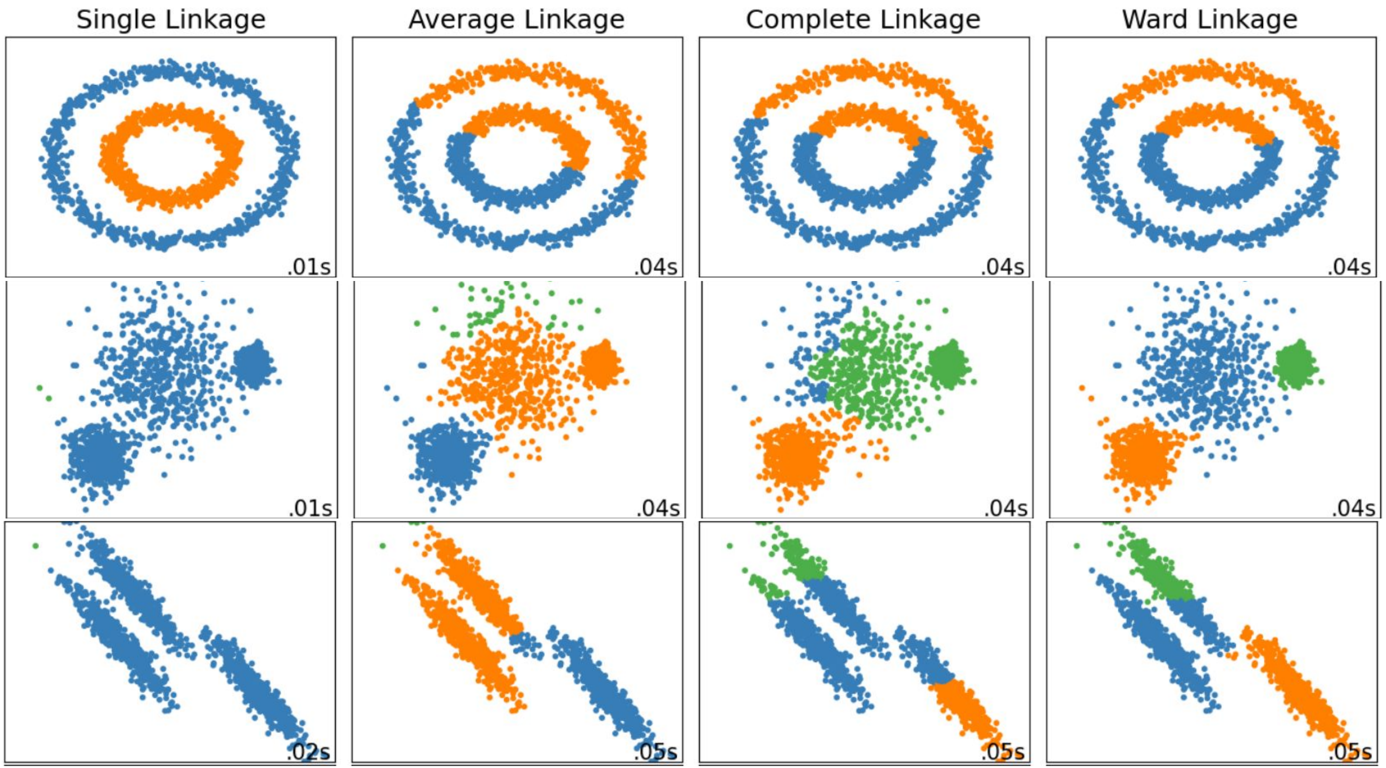
   - Examples: L^2, L^1, your favorite norm, $1 - cor(x, y)$

2. **Linkage metric:** rule for joining two clusters
   - Single linkage
   - Complete linkage
   - Average linkage
   - Ward's linkage



Linkage examples

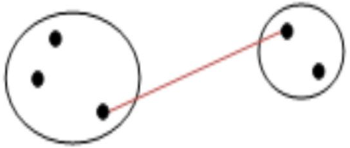




Linkages

Single Linkage (min)

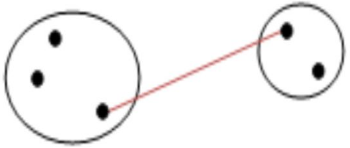
- $O(n^2)$
- Can handle diverse shapes
- Very sensitive to outliers or noise
- Often results in unbalanced clusters
- Extended, trailing clusters in which observations fused one at a time – chaining



Linkages

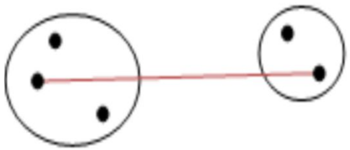
Single Linkage (min)

- $O(n^2)$
- Can handle diverse shapes
- Very sensitive to outliers or noise
- Often results in unbalanced clusters
- Extended, trailing clusters in which observations fused one at a time – chaining



Complete Linkage (max)

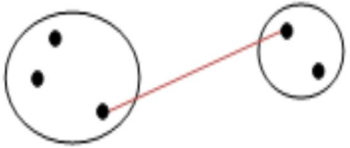
- $O(n^2)$
- Often gives cluster with similar sizes
- Less sensitive to outliers
- Works better with spherical distributions



Linkages

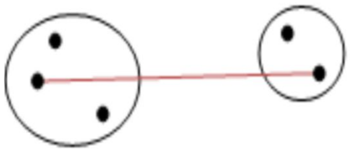
Single Linkage (min)

- $O(n^2)$
- Can handle diverse shapes
- Very sensitive to outliers or noise
- Often results in unbalanced clusters
- Extended, trailing clusters in which observations fused one at a time – chaining



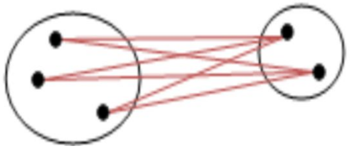
Complete Linkage (max)

- $O(n^2)$
- Often gives cluster with similar sizes
- Less sensitive to outliers
- Works better with spherical distributions



Average Linkage

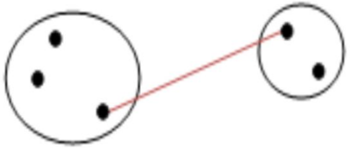
- A compromise between single and complete linkage
- Less sensitive to outliers than complete linkage, but not as robust as single linkage
- Works better with spherical distributions



Linkages

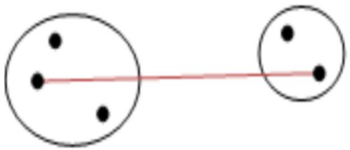
Single Linkage (min)

- $O(n^2)$
- Can handle diverse shapes
- Very sensitive to outliers or noise
- Often results in unbalanced clusters
- Extended, trailing clusters in which observations fused one at a time – chaining



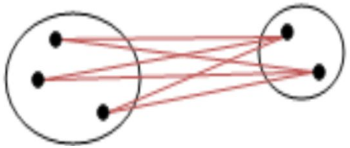
Complete Linkage (max)

- $O(n^2)$
- Often gives cluster with similar sizes
- Less sensitive to outliers
- Works better with spherical distributions



Average Linkage

- A compromise between single and complete linkage
- Less sensitive to outliers than complete linkage, but not as robust as single linkage
- Works better with spherical distributions



- **Ward's Linkage:** join sets that minimize the Euclidean distance between all pairs of points
- Average and Ward's linkages are most widely used

Hierarchical clustering

Advantages

- Gives nested family of clusterings
- Convenient visualizations with dendrograms

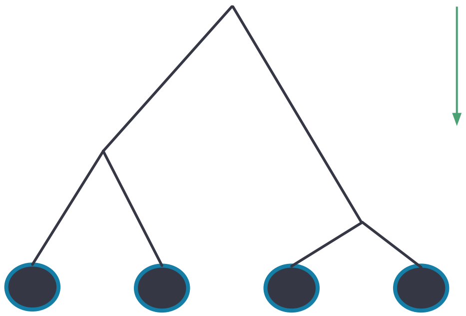
Disadvantages

- Depends heavily on linkage
- p >> n (“Curse of Dimensionality”)
- Local solution
- Slow: $O(n^3)$ in the worst case (depends on linkage)
- Uses a lot of memory: $O(n^2)$

Hierarchical clustering: divisive approach

Top-down approach ("divisive")

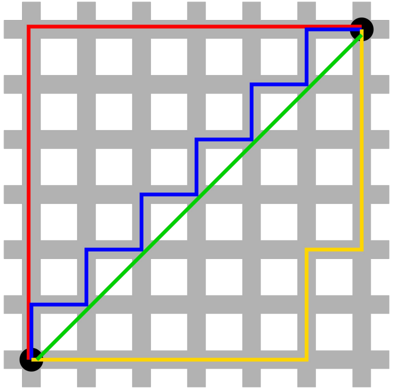
- Start with a single big cluster
- Use a subroutine (like k-means) to separate the data into successively more and more cluster
- **Advantages:**
  - Can be faster than agglomerative version (depends on subroutine)
  - Makes decisions based on global structure
- **Disadvantages:**
  - More complex, have to choose subroutine



k-Means (with L^2 distance)

Idea: find clusters C which minimize the within-cluster sum of squares

$$\underset{C}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} \|\mathbf{x} - \boldsymbol{\mu}_k\|_2^2, \quad \text{where} \quad \boldsymbol{\mu}_k = \frac{1}{n_k} \sum_{\mathbf{x} \in C_k} \mathbf{x}$$



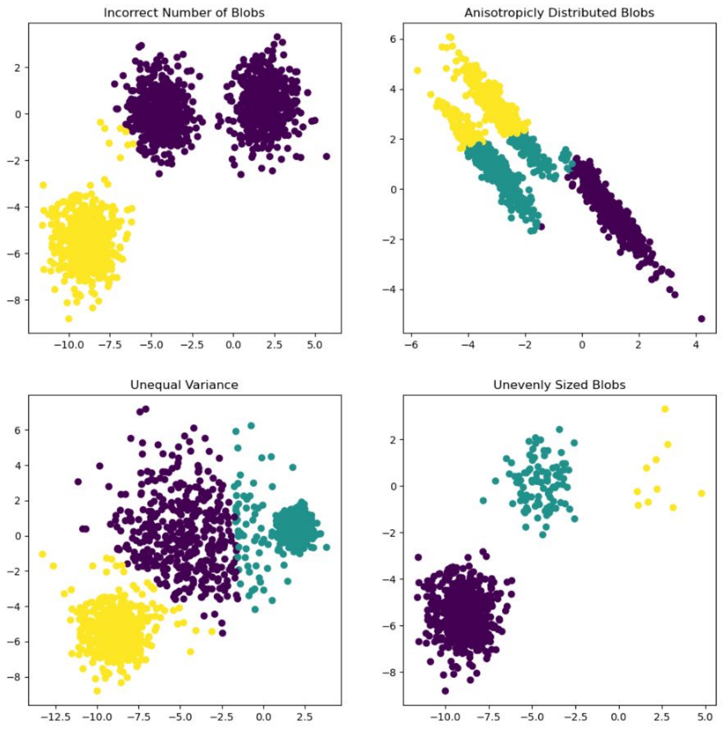
k-Means (with L^2 distance)

Advantages

- Fast
- Good when clusters are spherical balls and linearly separable

Disadvantages

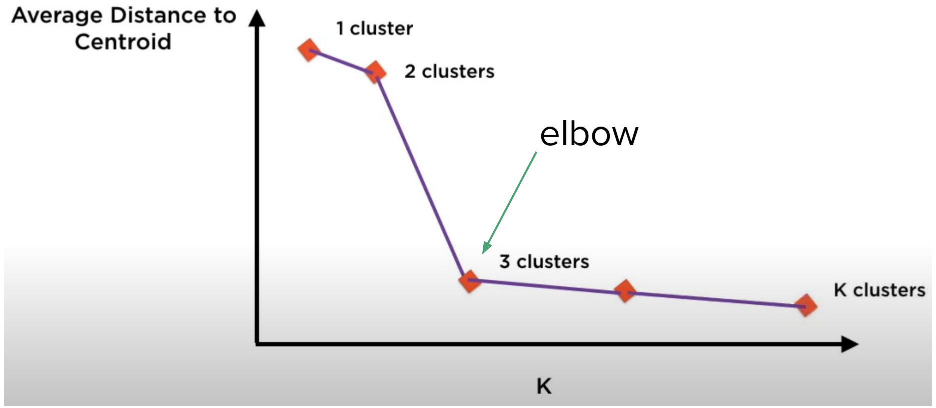
- Bad when clusters not spherical
- Bad when clusters have different variances
- p >> n (“Curse of Dimensionality”)
- Irrelevant variables are treated as equals with relevant ones
- Heuristic solution depends on initialization
  - Exact solution is NP-hard



Good discussion here: https://stats.stackexchange.com/questions/133656/how-to-understand-the-drawbacks-of-k-means

How to choose K?

- Elbow plots



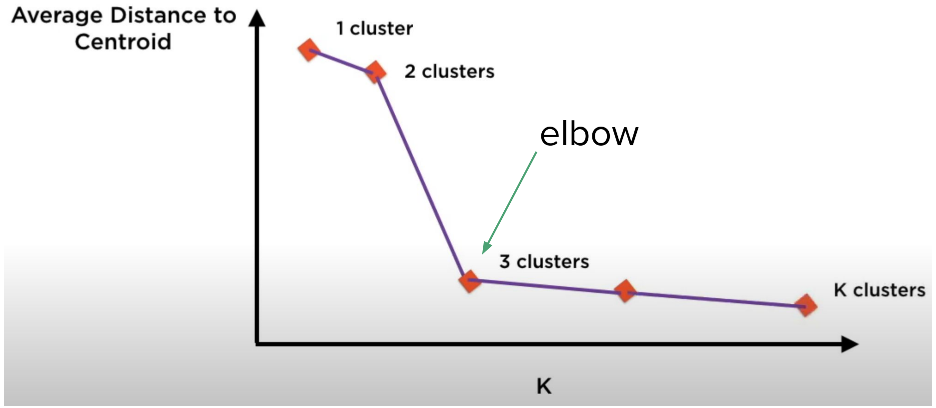
Source: https://www.youtube.com/watch?v=AtxQ0rvdQIA

How to choose K?

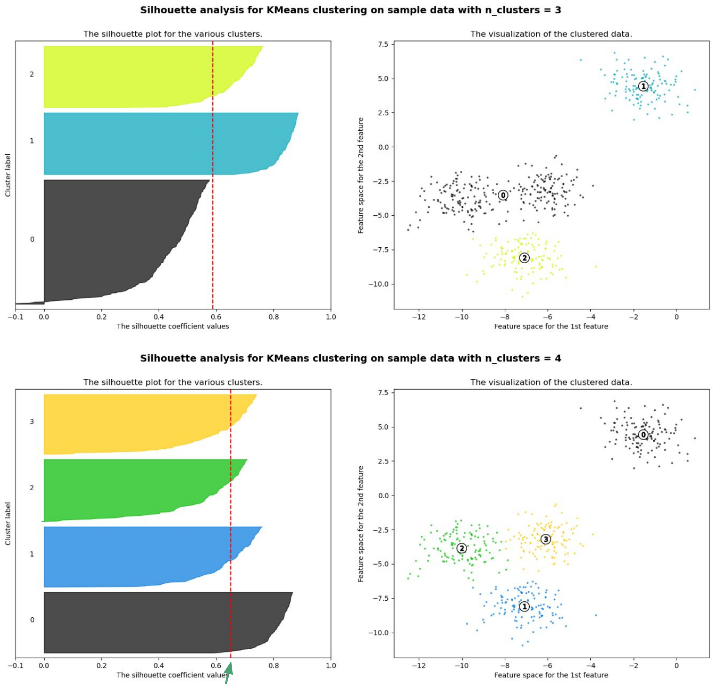
Source: [Selecting the number of clusters with silhouette analysis on KMeans clustering]

- Elbow plots
- Silhouette statistic. For each point, calculate:
  - avg. dist. to points in same cluster
  - avg. dist. to points in other clusters

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



Source: [https://www.youtube.com/watch?v=AtxQ0rvdQIA]



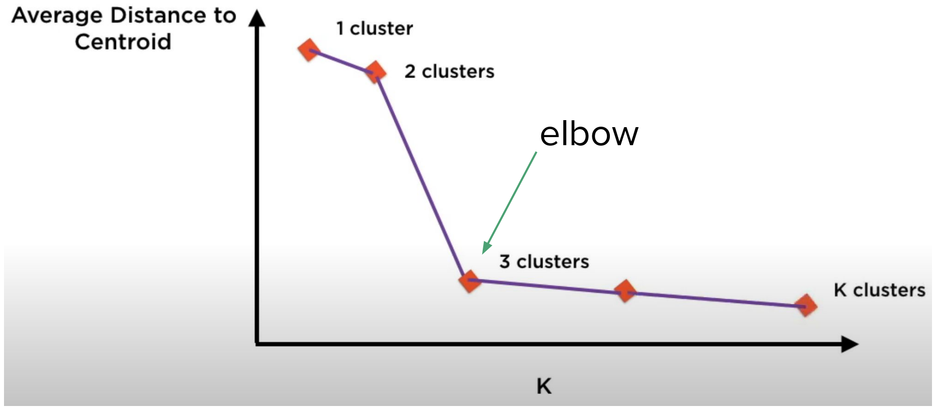
Choose K that leads to highest mean silhouette

How to choose K?

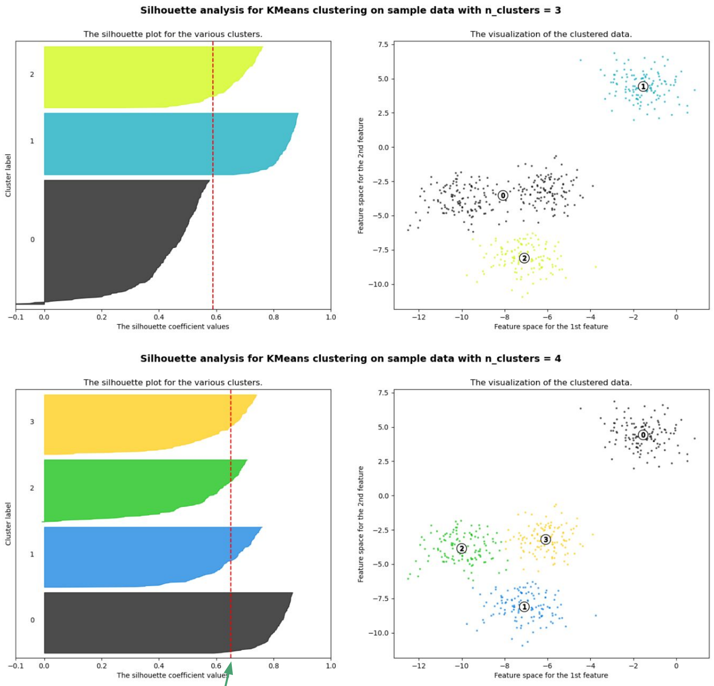
Source: [Selecting the number of clusters with silhouette analysis on KMeans clustering]()

- Elbow plots
- Silhouette statistic. For each point, calculate:
  - avg. dist. to points in same cluster
  - avg. dist. to points in other clusters
- Stability (will get some practice soon)

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



Source: https://www.youtube.com/watch?v=AtxQ0rvdQIA



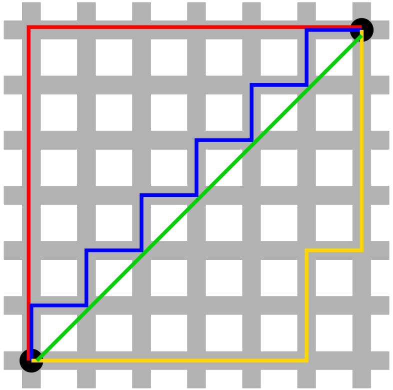
Choose K that leads to highest mean silhouette

k-Medians (with L^1 distance)

Idea: find clusters C which minimize the within-cluster absolute value

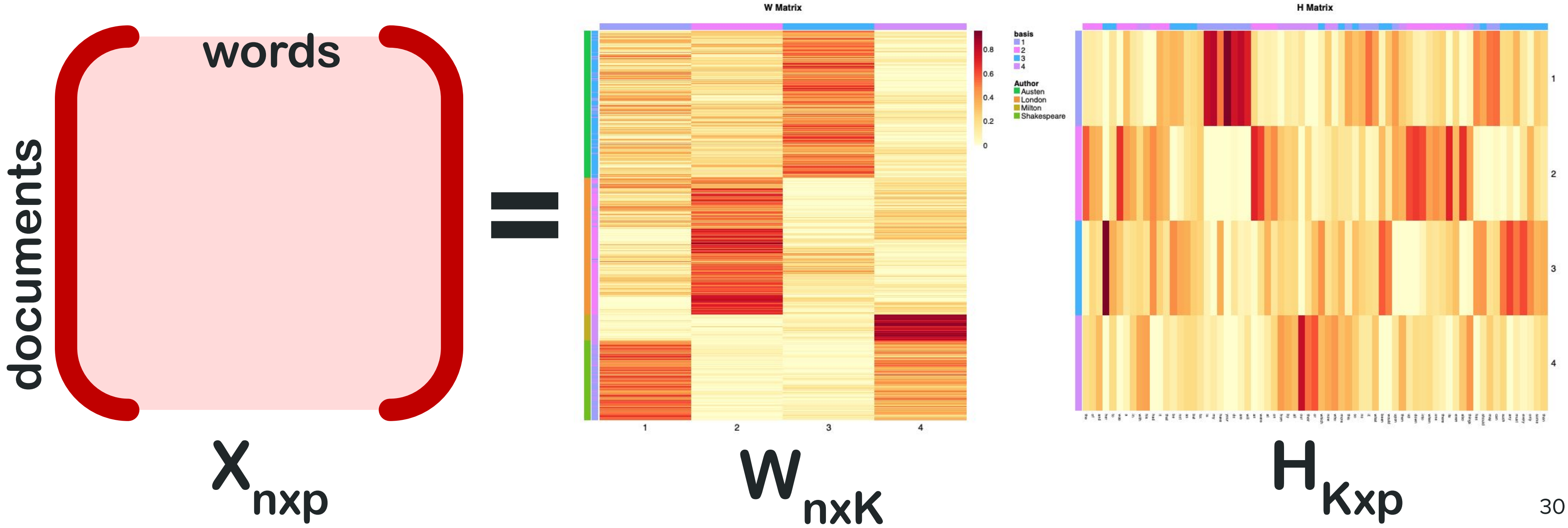
$$\mathrm{argmin}_C \sum_{k=1}^{K} \sum_{x \in C_k} \|x - m_k\|_1$$

vector of medians along each dimension



Nonnegative Matrix Factorizations (NMF)

- Given a non-negative matrix **X**, NMF solves $\min_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} \|\mathbf{X} - \mathbf{W}\,\mathbf{H}\|_F^2$
- Tool for dimension reduction, pattern recognition, and soft clustering with positive data



Nonnegative Matrix Factorizations (NMF)

Advantages

- “Soft” clustering
- Inherently gives sparse feature matrix **H** (unlike PCA)
- Great for pattern recognition with positive data

Disadvantages

- Not a convex problem depends on initialization of algorithm
- Components are unordered and not nested
  - Change number of components K can give vastly different results
- Can’t find *strength* of patterns as in PCA

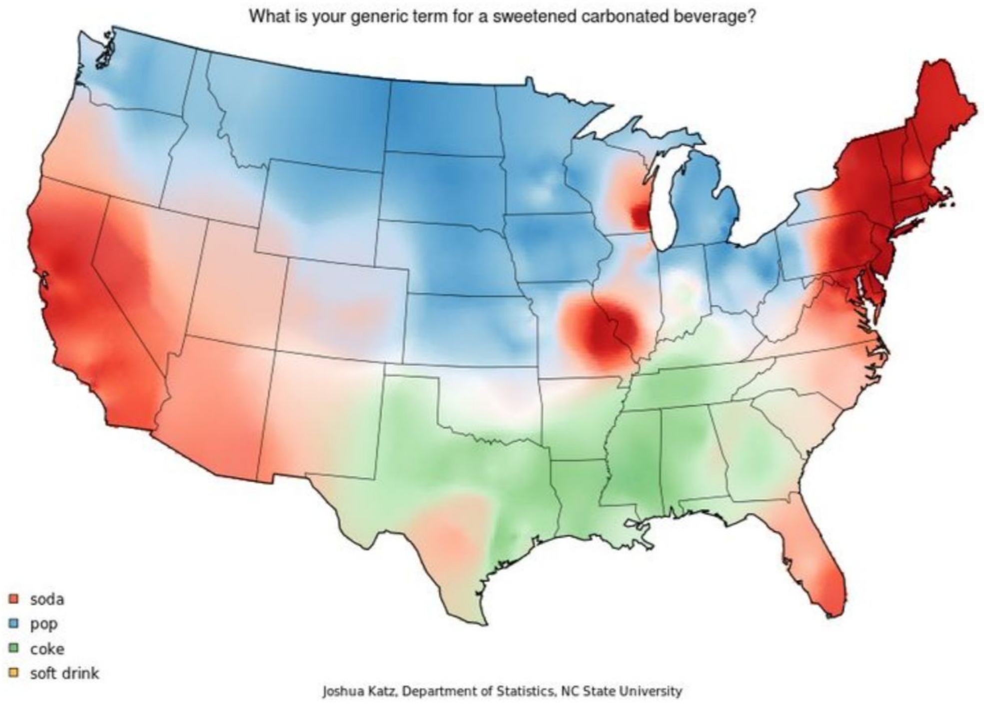
To center/scale or not to center/scale…

- By **centering**, I mean subtracting the sample mean from each column in your data matrix so that the mean of each column/feature is 0
- By **scaling**, I mean dividing each column by some constant so that the 2-norm of each column/feature is 1

- Very subjective…
- If it is meaningful to compare the variance of different features in your data matrix, don't *need* to scale
  - Gene expression data
- If features are measured on different scales, definitely need to scale (and maybe center?)
  - Income and number of kids
- Centering may result in a loss in interpretability (e.g., with positive data)
- What most people do in practice: try both

Go to
lab_week5/
folder and
work in groups

Lab 2 – Linguistics Survey (Due October 7 11:59pm)



https://www.businessinsider.com/22-maps-that-show-the-deepest-linguistic-conflicts-in-america-2013-6#ok-this-one-is-crazy-everyone-pronounces-pecan-pie-differently-10

Lab 2 – Linguistics Survey

- Your primary goal is to:
  - Perform EDA/dimension reduction and clustering
  - Evaluate stability of clustering
- Other things to keep in mind
  - Readability of narrative and code
  - Clear and effective visualizations
  - Clarity of folder structure
    - Only push files required to reproduce the report (minus data)
- Don’t forget about HW


KEEP
CALM
AND
START
EARLY