STAT 215A Fall 2019
Week 6

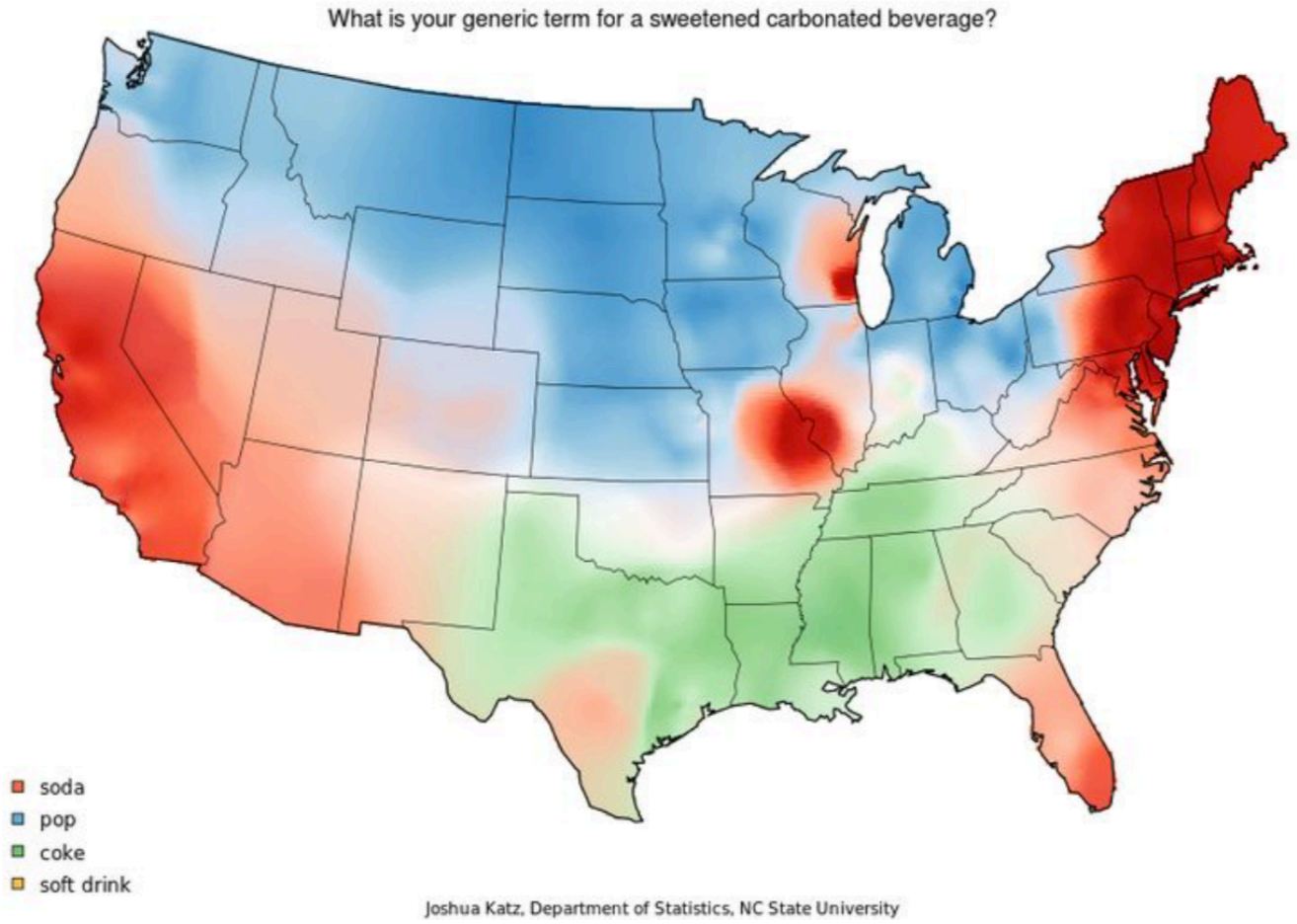
Tiffany Tang

10/4/19

Announcements

- Lab 1 grades have been pushed to your GitHub repos
  - Takeaways:
    - Be wary of overplotting – use smaller point sizes, transparency, subsampling
    - Think carefully about what you want to convey in your plot before deciding what how to make/present the plot
- Corrections in HW2
- Lab 2 due in one week: **October 10 11:59pm**

Questions on HW2 or Lab 2?



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

Some things to think about…

- Maps, maps, maps
- Regional differences in dialect
- Working with categorical variables
  - Normalization
- Missing data
- Project file structure
  - .gitignore

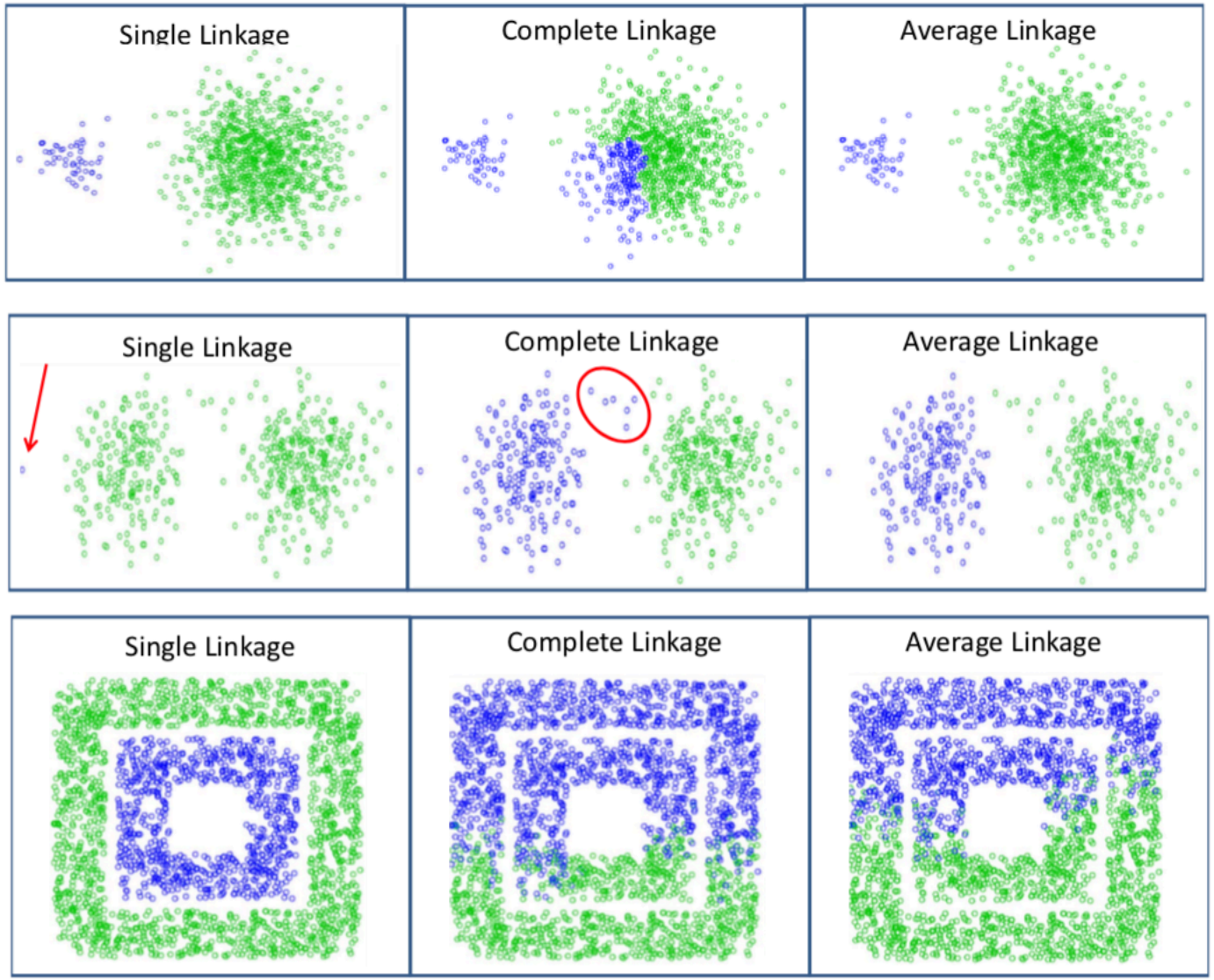
Plan for Today:

- Recap of the three clustering methods from last time
  - K-means
  - Hierarchical Clustering
  - NMF
- Clustering part II
  - DBScan
  - More details on how to choose K

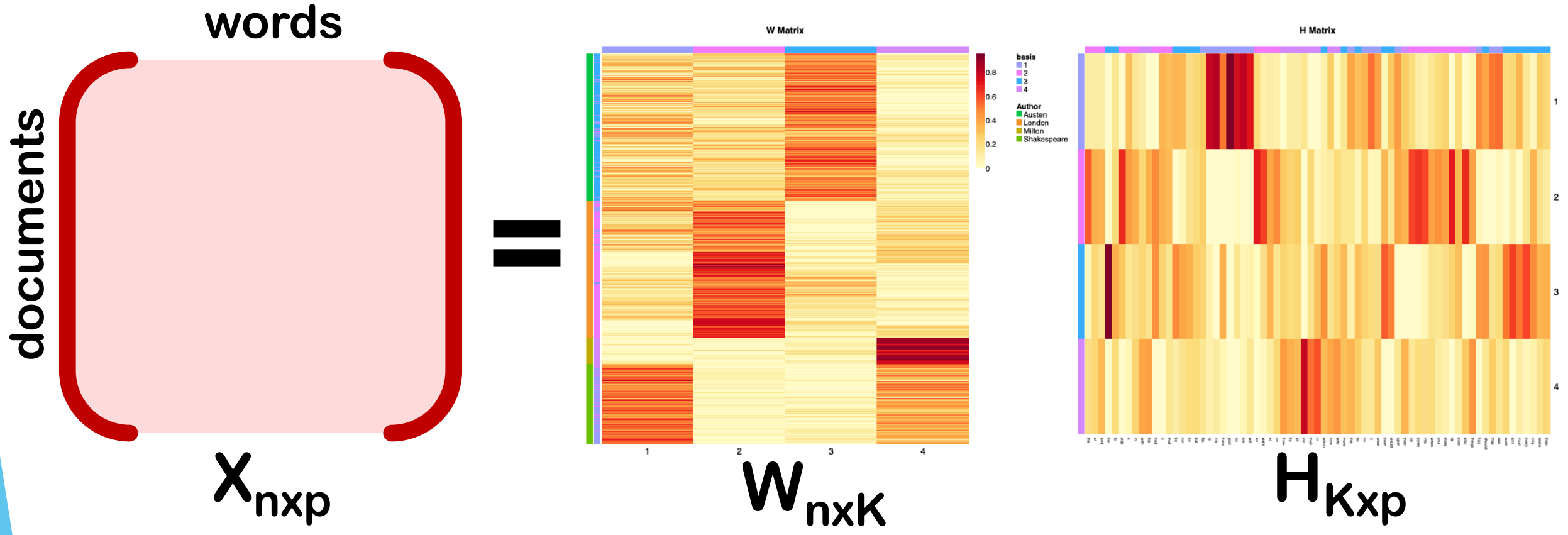
Review

	K-means	Hierarchical Clustering	NMF
Advantages	• Super fast and intuitive • Good when clusters are spherical balls and linearly separable	• Gives nested family of clusterings • Convenient visualizations with dendrograms	• Dimension reduction + soft clustering • Inherently gives sparse feature matrix H (unlike PCA) • Great with positive data
Disadvantages	• Bad when clusters not spherical or have different variances • Curse of dimensionality • Local solution	• Depends *heavily* on linkage (single, complete, average, Ward's linkage) • Curse of dimensionality • Local solution	• Components are unordered and not nested • Can't find *strength* of patterns as in PCA • Local solution • Doesn't make sense with negative data

Linkage Examples



Nonnegative Matrix Factorizations (NMF)

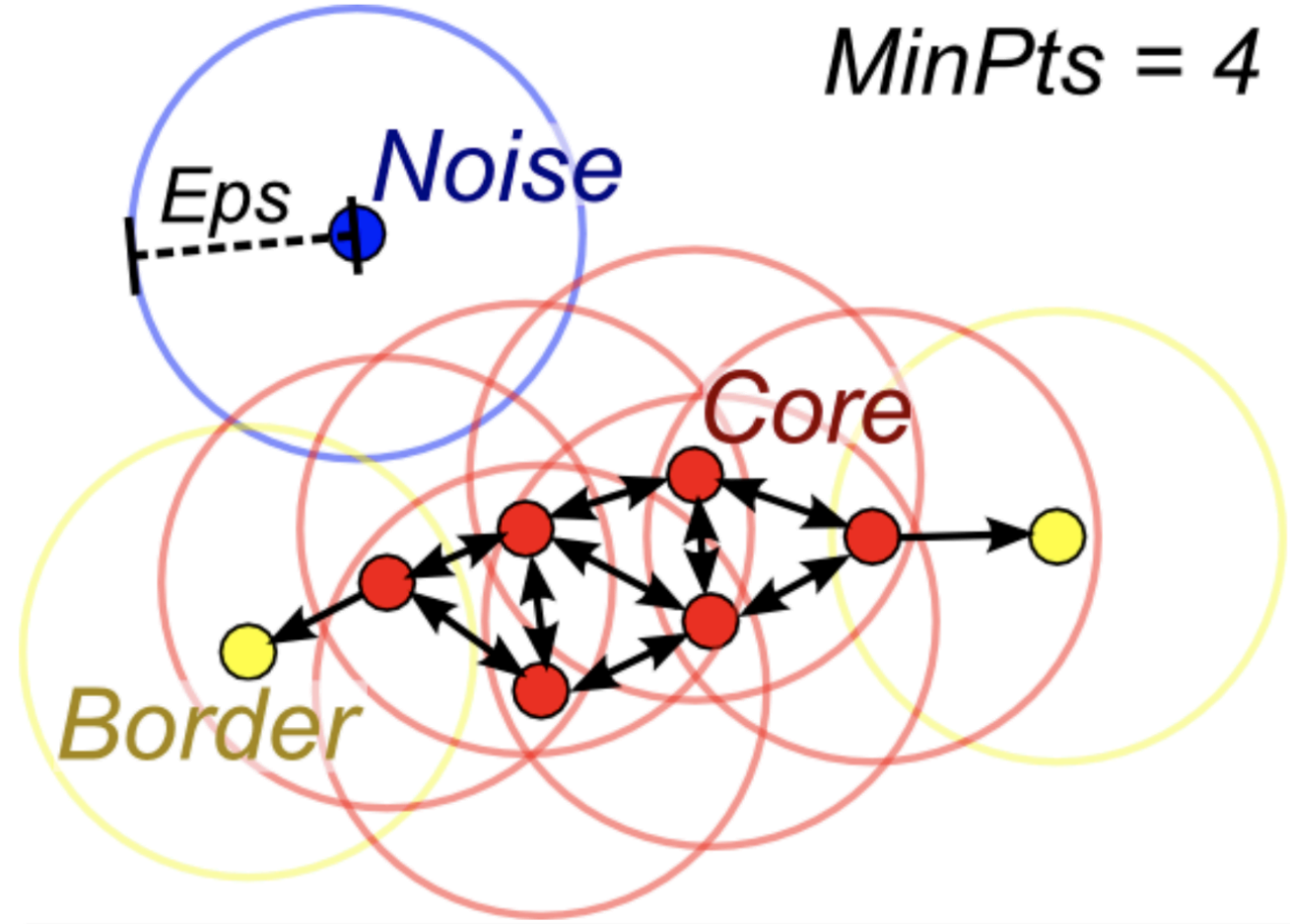


DBScan (Ester, Martin, et al. 1996)

- Density-based clustering
- **Idea:** group together points that are closely packed together (points with many nearby neighbors) while marking points that lie in low-density regions as outliers
- Choose (two?) parameters:
  - ε = how close points should be to each other to be considered a part of a cluster
    - Distance metric
  - minPts = minimum number of points required to form a dense region

Source: https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818

DBSCAN (Ester, Martin, et al. 1996)



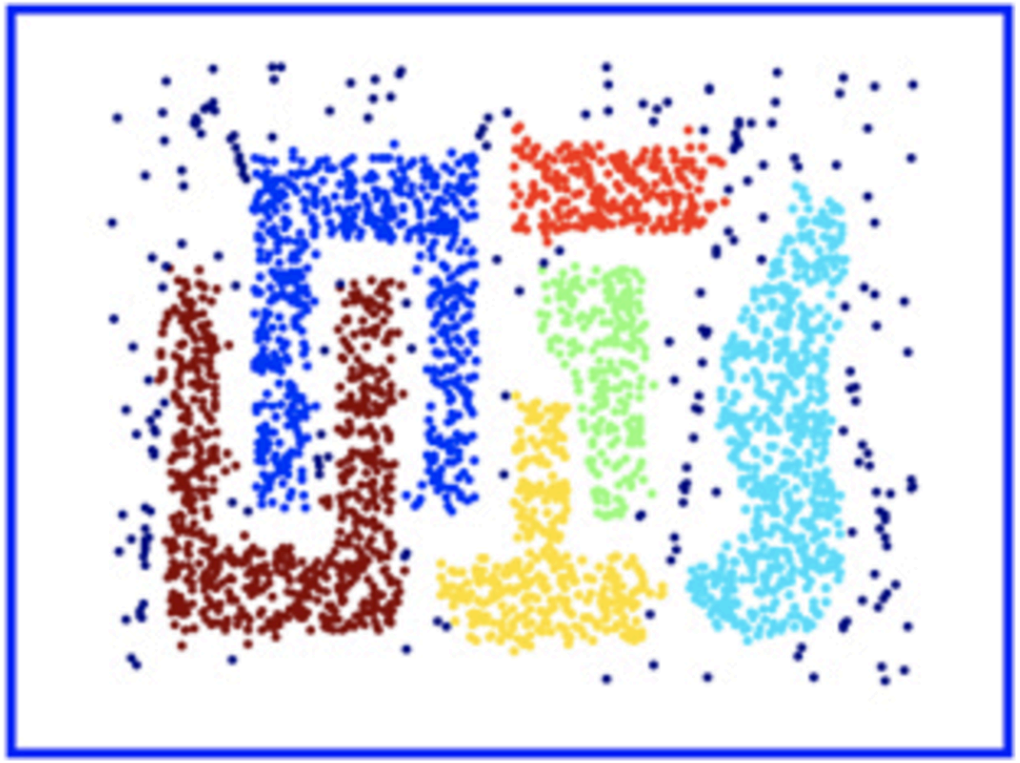
Red: Core Points

Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

Source: https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818

DBSCAN (Ester, Martin, et al. 1996)



In R: `dbscan::dbscan()`

Source: https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818

Quick Fun Example in R: President's data set

DBSCAN

Advantages

- Don’t have to choose K (though K depends on your choice of ε and minPts)
- Great for spatial data
- Great at separating clusters of similar densities that are well separated
- Robust to outliers
- Flexible to arbitrarily shaped clusters

Disadvantages

- If the data and scale are not well understood, choosing a meaningful distance threshold ε and minPts can be difficult
- Struggles when clusters are of varying densities since (ε, minPts) cannot then be chosen appropriately for all clusters
- Curse of dimensionality when distance metric = Euclidean distance
- Algorithm depends on ordering of points: border points that are reachable from more than one cluster can be part of either cluster, depending on the order the data are processed

K-Means

DBSCAN

Hierarchical Clustering

NMF

and more...

How to choose K/other tuning parameters

- This is an open area of research!
- Method-specific tools:
  - K-means and hierarchical clustering: silhouette method
  - NMF: cross-validation/missing data imputation
- A more general idea/tool: stability with respect to…
  - Data perturbations (e.g., bootstrap, subsampling)
  - Algorithmic perturbations (e.g., random initializations)

How to choose K/other tuning parameters

- This is an open area of research!
- Method-specific tools:
  - **K-means and hierarchical clustering: silhouette method**
  - NMF: cross-validation/missing data imputation
- A more general idea/tool: stability with respect to…
  - Data perturbations (e.g., bootstrap, subsampling)
  - Algorithmic perturbations (e.g., random initializations)

Silhouette Method

- A “good” cluster is one where points within the same cluster are close together and points in different clusters are far away from each other
- Silhouette Metric:

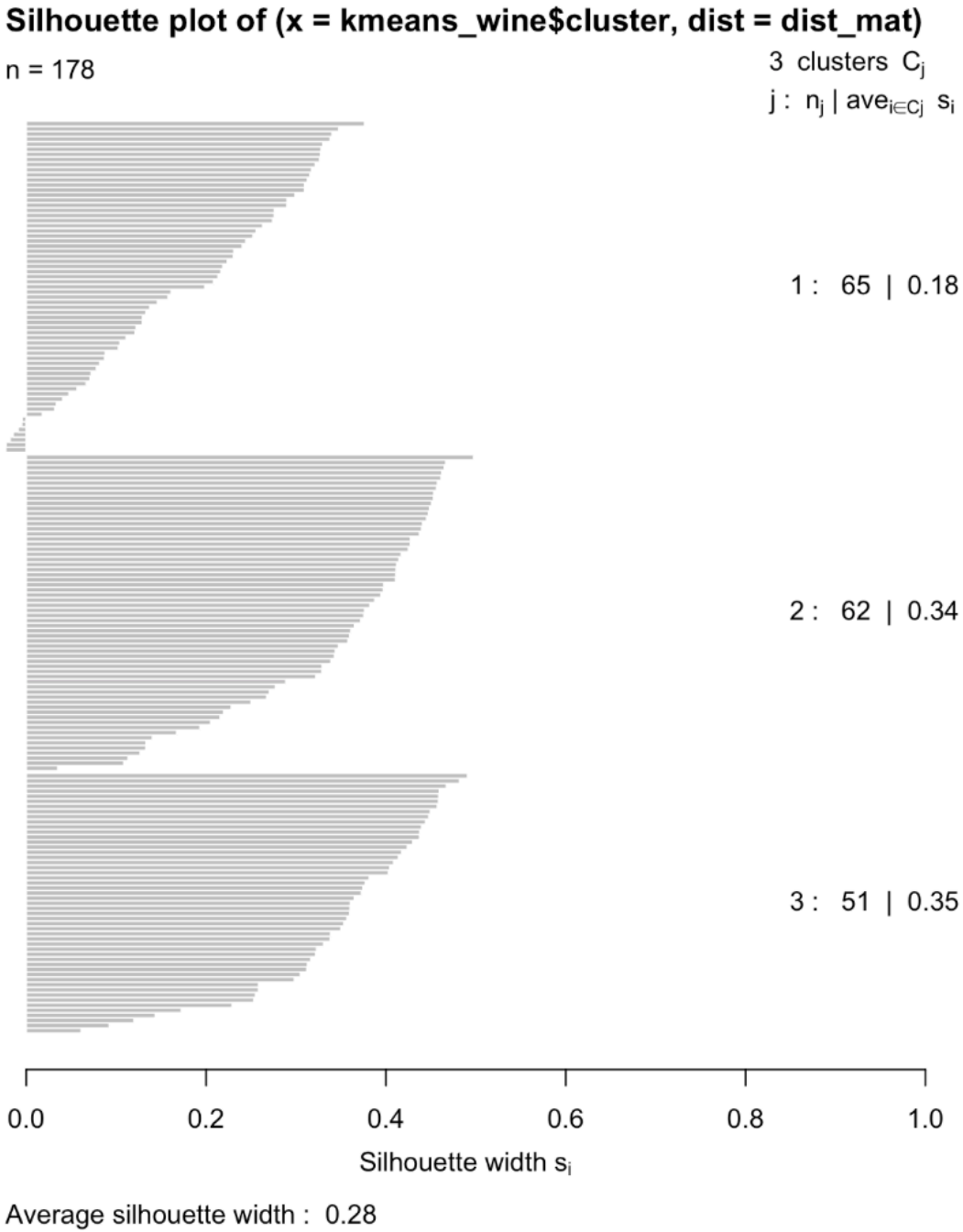
$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

where $a_i = \sum_{j \in C_i} d(x_i, x_j)$ is the average distance between x_i and all other points belonging to the same cluster C_i and $b_i = \min_{C_k} \sum_{j \in C_k} d(x_i, x_j)$ for $k \neq i$ is the average dissimilarity between x_i and the nearest cluster to which x_i does not belong.

- Want silhouette metric to be large
- Note: you can compute this metric given any vector of cluster memberships; clusters can be from k-means, hierarchical clustering, etc.
- But mostly used for k-means and hierarchical clustering

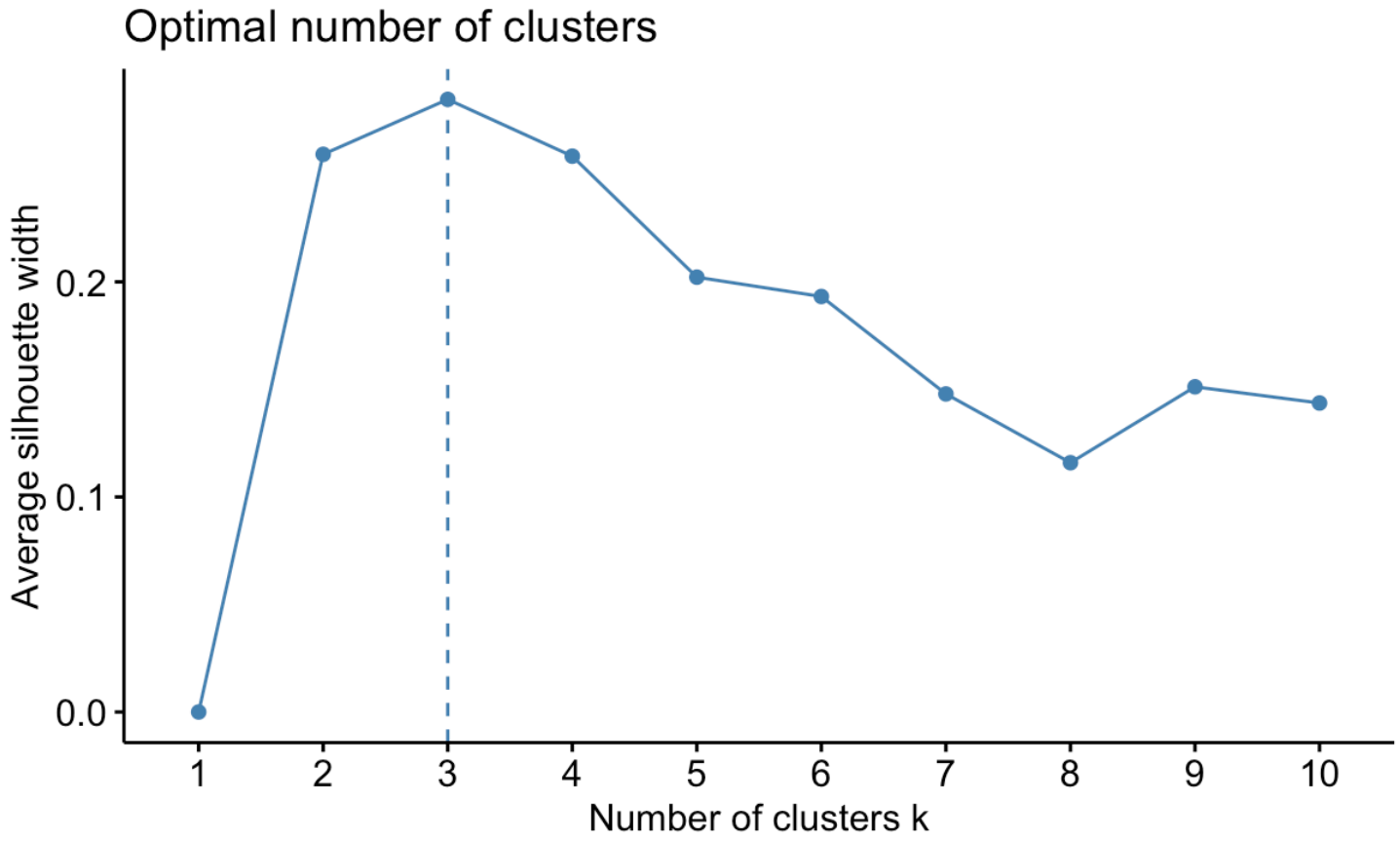
Silhouette Method

- For a fixed choice of K, can plot the silhouette score for each observation
- See the silhouette() function from the cluster package in R
- Can also look at these plots for various choices of K



Silhouette Method

- How to compare different choices of K?
- Compute the average silhouette width for each choice of K
- Pick the K with the highest average silhouette width
- See fviz_nbclust() function in the factoextra package in R



How to choose K/other tuning parameters

- This is an open area of research!
- Method-specific tools:
  - K-means and hierarchical clustering: silhouette method
  - **NMF: cross-validation/missing data imputation**
- A more general idea/tool: stability with respect to…
  - Data perturbations (e.g., bootstrap, subsampling)
  - Algorithmic perturbations (e.g., random initializations)

Nonnegative Matrix Factorizations (NMF)

- Given a non-negative matrix **X**, NMF solves

$$\underset{\mathbf{W}\geq 0,\mathbf{H}\geq 0}{\operatorname{argmin}} \; || \mathbf{X} - \mathbf{W}\,\mathbf{H} ||_F^2 = \sum_{i,j} (\mathbf{X}_{ij} - \mathbf{W}_i^\top \mathbf{H}_j)^2$$

- Turns out you can modify this optimization problem to work for **X** with missing data (see `NNLM::nnmf()` in R):

$$\underset{\mathbf{W}\geq 0,\mathbf{H}\geq 0}{\operatorname{argmin}} \sum_{\substack{(i,j) \\ \text{not missing}}} (\mathbf{X}_{ij} - \mathbf{W}_i^\top \mathbf{H}_j)^2$$

- **Idea for choosing K:**
  - Randomly leave out scattered missing elements of the data **X**
  - For each potential rank k,
    1. Apply NMF to the data with the missing values → $\mathbf{W}_m$ and $\mathbf{H}_m$
    2. Impute the missing values of **X** by $\mathbf{W}_m\mathbf{H}_m$
    3. Compute the imputation error (i.e., the mean squared difference between the imputed values and the observed values)
  - Can repeat this many times to get a mean and SE for each potential rank k
  - Similar to CV, choose the k by taking the minimum or using the 1-SE rule

How to choose K/other tuning parameters

- This is an open area of research!
- Method-specific tools:
  - K-means and hierarchical clustering: silhouette method
  - NMF: cross-validation/missing data imputation
- **A more general idea/tool: stability with respect to…**
  - **Data perturbations (e.g., bootstrap, subsampling)**
  - **Algorithmic perturbations (e.g., random initializations)**

Stability

- **Data perturbations:**
  - For each bootstrap:

    Run the method on the bootstrapped data

    Get the clusters
  - Compare the clusters from various perturbations
- **Algorithmic perturbations**
  - For trial in 1:ntrials:

    Run the method with a different initializations

    Get the clusters
  - Compare the clusters from various perturbations
- For now, compare visually or heuristically
  - Confusion matrices via `table(cluster1, cluster2)` in R
- Stay tuned for lab 3

In Class Labs

- **Week1:** tidyverse basics
- **Week2:** ggplot + Rmd tips and tricks
- **Week3:** more ggplot + additional plotting tools (pair plots, heatmaps, etc)
- **Week4:** PCA
- **Week5:** Kmeans, hierarchical clustering, NMF