MMAI 5040 – Winter 2022
Business Application of AI 1

Session 5: Cluster Analysis

February 7, 2022

Divinus Oppong-Tawiah, PhD.

Today's Class …

1. Announcements

2. Clustering
   - Main Idea
   - Measuring Cluster Distances
   - Hierarchical Clustering
   - Non-hierarchical clustering
   - Validating Clusters

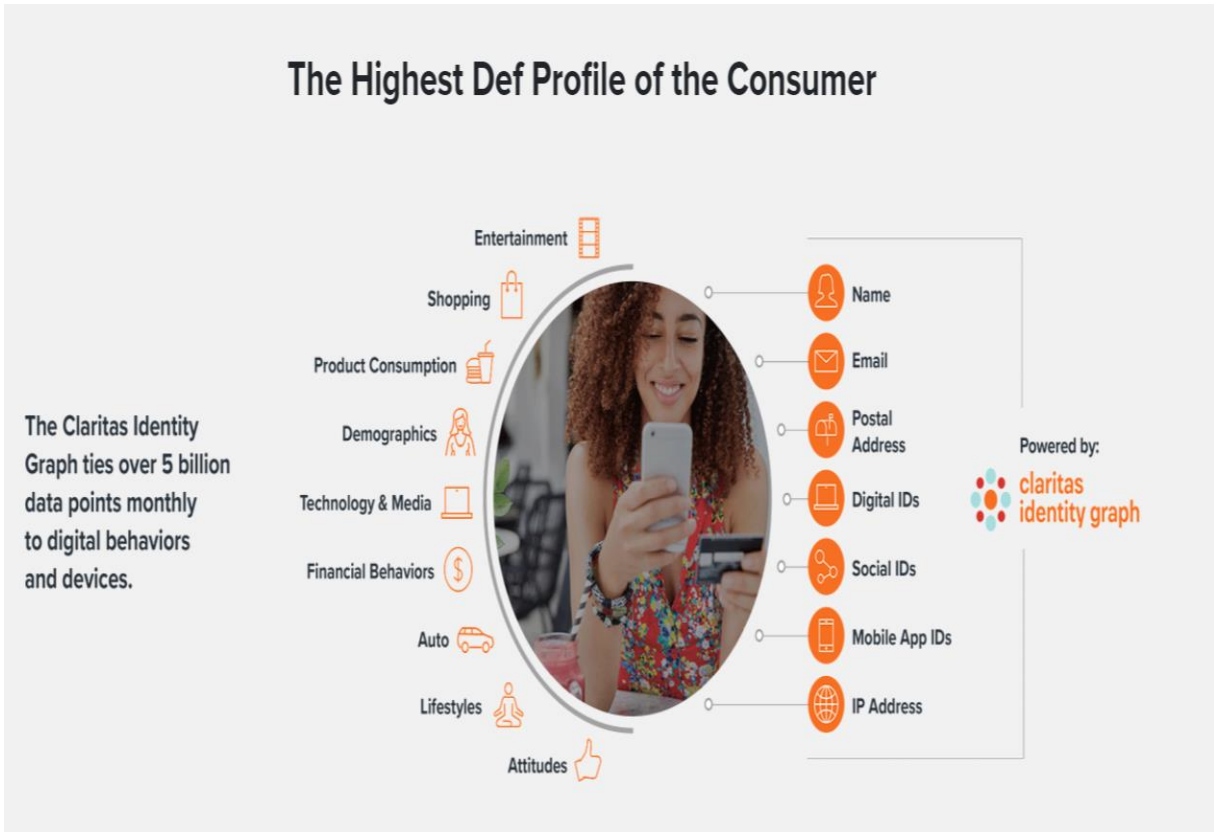
3. Practice Lab

CLUSTERING

Clustering: The Main Idea

CLARITAS PRIZM Cluster Profiles

https://youtu.be/7z5Wqt9kFVo



https://claritas.com/claritas-identity-graph/

Clustering: The Main Idea

- Goal: Form groups (clusters) of similar records

- Prominent business applications:
  - **Market segmentation**: identifying groups of similar customers e.g., Gen X, Gen Y, etc.
  - **Market structure analysis**: identifying groups of similar products, e.g., neighborhood clusters by lifestyle

Other Applications

- Grouping securities to create balanced portfolios
- Periodic table of the elements
- Classification of species
- Grouping firms for structural analysis of economy
- Army uniform sizes
- Segments of voters
- …

Example: Public Utilities

Goal: find clusters of similar utilities

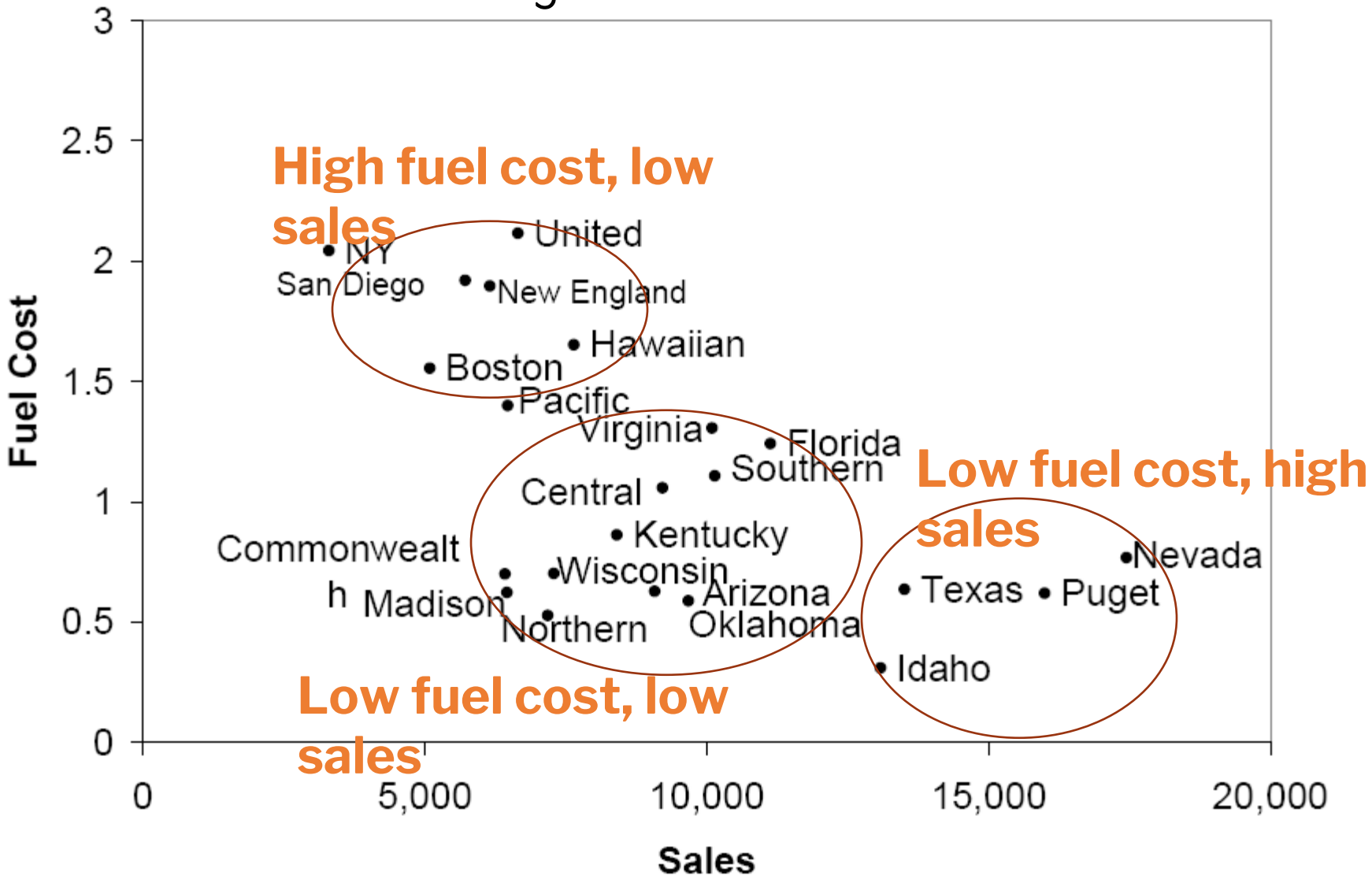
Data: 22 firms, 8 variables

- -Fixed-charge covering ratio
- -Rate of return on capital
- -Cost per kilowatt capacity
- -Annual load factor
- -Growth in peak demand
- -Sales
- -% nuclear
- -Fuel costs per kwh

Company	Fixed_charge	RoR	Cost	Load	Δ Demand	Sales	Nuclear	Fuel_Cost
Arizona	1.06	9.2	151	54.4	1.6	9077	0	0.628
Boston	0.89	10.3	202	57.9	2.2	5088	25.3	1.555
Central	1.43	15.4	113	53	3.4	9212	0	1.058
Commonwealth	1.02	11.2	168	56	0.3	6423	34.3	0.7
Con Ed NY	1.49	8.8	192	51.2	1	3300	15.6	2.044
Florida	1.32	13.5	111	60	-2.2	11127	22.5	1.241
Hawaiian	1.22	12.2	175	67.6	2.2	7642	0	1.652
Idaho	1.1	9.2	245	57	3.3	13082	0	0.309
Kentucky	1.34	13	168	60.4	7.2	8406	0	0.862
Madison	1.12	12.4	197	53	2.7	6455	39.2	0.623
Nevada	0.75	7.5	173	51.5	6.5	17441	0	0.768
New England	1.13	10.9	178	62	3.7	6154	0	1.897
Northern	1.15	12.7	199	53.7	6.4	7179	50.2	0.527
Oklahoma	1.09	12	96	49.8	1.4	9673	0	0.588
Pacific	0.96	7.6	164	62.2	-0.1	6468	0.9	1.4
Puget	1.16	9.9	252	56	9.2	15991	0	0.62
San Diego	0.76	6.4	136	61.9	9	5714	8.3	1.92
Southern	1.05	12.6	150	56.7	2.7	10140	0	1.108
Texas	1.16	11.7	104	54	-2.1	13507	0	0.636
Wisconsin	1.2	11.8	148	59.9	3.5	7287	41.1	0.702
United	1.04	8.6	204	61	3.5	6650	0	2.116
Virginia	1.07	9.3	174	54.3	5.9	10093	26.6	1.306

2D Scatter plot of Sales & Fuel Cost:

3 rough clusters can be seen



Extension to More Than 2 Dimensions

- In prior example, clustering was done by eye
- Multiple dimensions require formal algorithm with
  - A **distance measure**
  - A way to use the distance measure in forming clusters
- We will consider two algorithms:
  - **hierarchical (agglomerative)** and
  - **non-hierarchical (k-means)**

Measuring Distances

- Between records
- Between clusters

Distance Between Two Records

Let d_{ij} be a distance similarity metric between records *i, j* :

$$i = [x_{i1}, x_{i2}, \dots, x_{ip}]$$

$$j = [x_{j1}, x_{j2}, \dots, x_{jp}]$$

Multiple distance measures can be defined follow these general properties:

$$Non\_negativity: \quad d_{ij} \geq 0$$

$$Self\_proximity: \quad d_{ii} = 0$$

$$Symmetry: \quad d_{ij} = d_{ji}$$

$$Triangle\ inequality: \quad d_{ij} \leq d_{ik} + d_{kj}$$

Distance Between Two Records

Euclidean Distance is most popular:

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

Problem: Raw distance measures are highly influenced by scale of measurements

Solution:

- **Normalize (standardize) the data**
  - ❑ Subtract mean, divide by std. deviation
  - ❑ Also called **z-scores**
- **Or Rescale the data to [0,1] range**
  - ❑ Subtract the minimum
  - ❑ Divide by range

Compute Distance Matrix for Utility Pairs

Partial Output

Company	Arizona	Boston	Central	Commonwealth	NY
Arizona	0.000000				
Boston	3989.408076	0.000000			
Central	140.402855	4125.044132	0.000000		
Commonwealth	2654.277632	1335.466502	2789.759674	0.000000	
NY	5777.167672	1788.068027	5912.552908	3123.153215	0.000000

E.g., For 22 utilities: Avg. Sales = 8,914; Std. dev. = 3,550

Normalized score for Arizona Sales: (9,077-8,914)/3,550 = 0.046

Normalized Distance Matrix

Partial output

Company	Arizona	Boston	Central	Commonwealth	NY
Arizona	0.000000				
Boston	2.010329	0.000000			
Central	0.774179	1.465703	0.000000		
Commonwealth	0.758738	1.582821	1.015710	0.000000	
NY	3.021907	1.013370	2.432528	2.571969	0.000000

Distance Between Two Records

Limitations of Euclidean Distance

- Highly scale dependent (e.g., cents vs dollars)
  - Common solution (Normalization) does not allow unequal weighting of variables when desired
- Completely ignores the relationship between variables (e.g., strong correlations)
- Sensitivity to outliers

Other Distance Measures

- Correlation-based similarity: $r_{ij} = \frac{\sum_{m=1}^{p}(x_{im}-\bar{x}_m)(x_{jm}-\bar{x}_m)}{\sqrt{\sum_{m=1}^{p}(x_{im}-\bar{x}_m)^2}\sqrt{\sum_{m=1}^{p}(x_{jm}-\bar{x}_m)^2}}$

  $d_{ij} = r_{ij}^2\ (similarity)$

  $d_{ij} = 1 - r_{ij}^2\ (dissimilarity\ or\ distance)$
- Statistical distance (Mahalanobis):

  $$d_{ij} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)' S^{-1} (\mathbf{x}_i - \mathbf{x}_j)}$$

  where $\mathbf{x}_i$ and $\mathbf{x}_j$ are p-dimensional vectors and S is their covariance matrix
- Manhattan distance (Absolute difference):

  $$d_{ij} = \sum_{m=1}^{p} |x_{im} - x_{jm}|$$
- Maximum coordinate distance:

  $$d_{ij} = \max_{m=1,2,\ldots,p} |x_{im} - x_{jm}|$$
- Gower's similarity (for mixed variable types): (continuous & categorical):

  $$d_{ij} = \frac{\sum_{m=1}^{p} w_{ijm} s_{ijm}}{\sum_{m=1}^{p} w_{ijm}}$$

  where s_{ijm} and w_{ijm} are weights defined for continuous and categorical measures

For Categorical Data: Similarity

- Use similarity, rather than distance, for categorical data
- To measure the distance between records *i, j* in terms of two 0/1 variables, create table with counts:

	0	1
0	a	b
1	c	d

Similarity metrics based on this table:

- Matching coef. = (a+d)/p
- Jaquard's coef. = d/(b+c+d)
  - for cases where a matching "1" is much greater evidence of similarity than matching "0" (e.g., ?)

Choosing a Distance Measure

Five important, domain-specific questions to consider:

- What exactly is being measured?
- How are the different measures related?
- On what scale is each measurement (numerical, ordinal, or nominal)?
- Are there outliers?
- Should measures be evenly or unevenly weighted?

Distance Between Two Clusters

- A cluster is a set of one or more records
- Let d_{AB} be the distance between two clusters *A* and *B* with *m* and *n* records respectively s.t.

$$A = [A_1, A_2, \dots, A_n]$$

$$B = [B_1, B_2, \dots, B_n]$$

- Most common cluster distances are:
  1. Minimum distance
  2. Maximum distance
  3. Average distance
  4. Centroid distance

1. <u>Minimum Distance (Cluster A to Cluster B)</u>
   - Also called **single linkage**
   - Distance between two clusters is the distance between the pair of records A_i and B_j that are closest

$$d_{AB} = \min_{i=1,2,\dots,m;\quad j=1,2,\dots,n,} \left(\text{distance}(A_i, B_j)\right),$$

2. <u>Maximum Distance</u>
   - Also called **complete linkage**
   - Distance between two clusters is the distance between the pair of records A_i and B_j that are farthest from each other

$$d_{AB} = \max_{i=1,2,\dots,m;\quad j=1,2,\dots,n,} \left(\text{distance}(A_i, B_j)\right),$$

Distance Between Two Clusters

- A cluster is a set of one or more records
- Let d_{AB} be the distance between two clusters *A* and *B* with *m* and *n* records respectively s.t.

$$A = [A_1, A_2, \dots, A_n]$$

$$B = [B_1, B_2, \dots, B_n]$$

- Most common cluster distances are:
  1. Minimum distance
  2. Maximum distance
  3. Average distance
  4. Centroid distance

3. <u>Average Distance</u>
   - Also called **average linkage**
   - Distance between two clusters is the average of all possible pair-wise distances

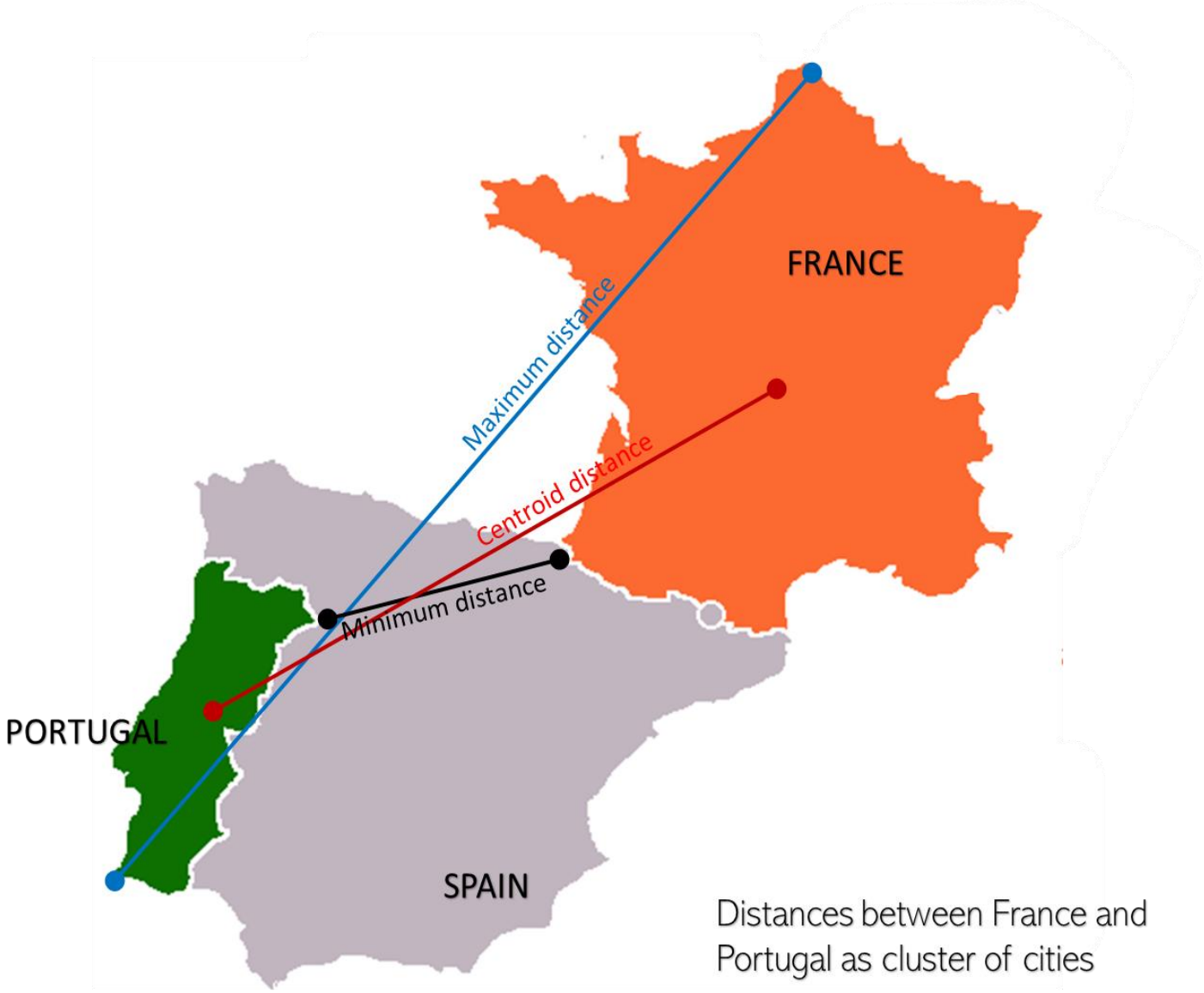
$$d_{AB} = \text{Average}\left(\text{distance}(A_i, B_j)\right), \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n,$$

4. <u>Centroid Distance</u>
   - Centroid is the vector of variable averages for all records
   - Distance between two clusters is the distance between the two cluster centroids

$$Cluster\ centroid: \bar{x}_A = \left[\left(\frac{1}{m}\sum_{i=1}^{m} x_{1i}, \dots, \frac{1}{m}\sum_{i=1}^{m} x_{pi}\right),\right]_{\substack{m = no.of\ measures,\\ p = no.of\ records}}$$

$$Centroid\ distance: d_{AB} = \text{distance}(\bar{x}_A, \bar{x}_B)$$

Distance Between Two Clusters



Distances between France and Portugal as cluster of cities

Question:

When and why would you choose each cluster distance:

- Minimum : for chain-like cluster shapes
- Maximum : for spherical cluster shapes / default choice if shape is unknown
- Average: same as Maximum distance

Calculating Distances Between Clusters

Original & Normalized Measurements

	Company	Sales	Fuel	*NormSales*	*NormFuel*
Cluster A	Arizona	9077	0.628	0.0459	-0.8537
	Boston	5088	1.555	-1.0778	0.8133
Cluster B	Central	9212	1.058	0.083	-0.0804
	Commonwealth	6432	0.7	-0.7017	-0.7242
	NY	3300	2.044	-1.5814	1.6926

Distance Matrix - *Normalized*

	Company	Arizona	Boston	Central	C'wealth	NY
Cluster A	Arizona	0.0000				
	Boston	2.0100	0.0000			
Cluster B	Central	0.7742	1.4657	0.0000		
	Commonwealth	0.7587	1.5828	1.0157	0.0000	
	NY	3.0219	1.0133	2.4325	2.5719	0.0000

Examine the distance matrix:

1. Closest pair is Arizona and Commonwealth
   - Minimum distance btn clusters A & B is 0.7587

2. Farthest pair is Arizona and NY
   - Maximum distance is 3.0219

3. Average distance btn clusters is:
   - (0.7742 + 0.7587 + 3.0219 + 1.4657 + 1.5838 + 1.0133) / 6 = 1.4362

4. Centroid distance btn clusters is:

$$= \sqrt{(-0.516 + 0.733)^2 + (-0.02 - 0.296)^2}$$

$$= \mathbf{0.3833}$$

Examine the normalized measurements:

$$Centroid\ of\ cluster\ A = \left[\frac{0.0459-1.0778}{2}, \frac{-0.8537+0.8133}{2}\right] = [-0.516, -0.02\ ]$$

$$Centroid\ of\ cluster\ B = \left[\frac{0.0839-0.7017-1.5814}{3}, \frac{-0.0804-0.7242+1.6926}{3}\right] = [-0.733, 0.296\ ]$$

10 minutes break

Hierarchical Clustering

Hierarchical Clustering

1. Agglomerative Methods

- Begin with n-clusters (each record its own cluster)
- Keep joining records into clusters until one cluster is left (the entire data set)
- Most popular

2. Divisive Methods

- Start with one all-inclusive cluster
- Repeatedly divide into smaller clusters

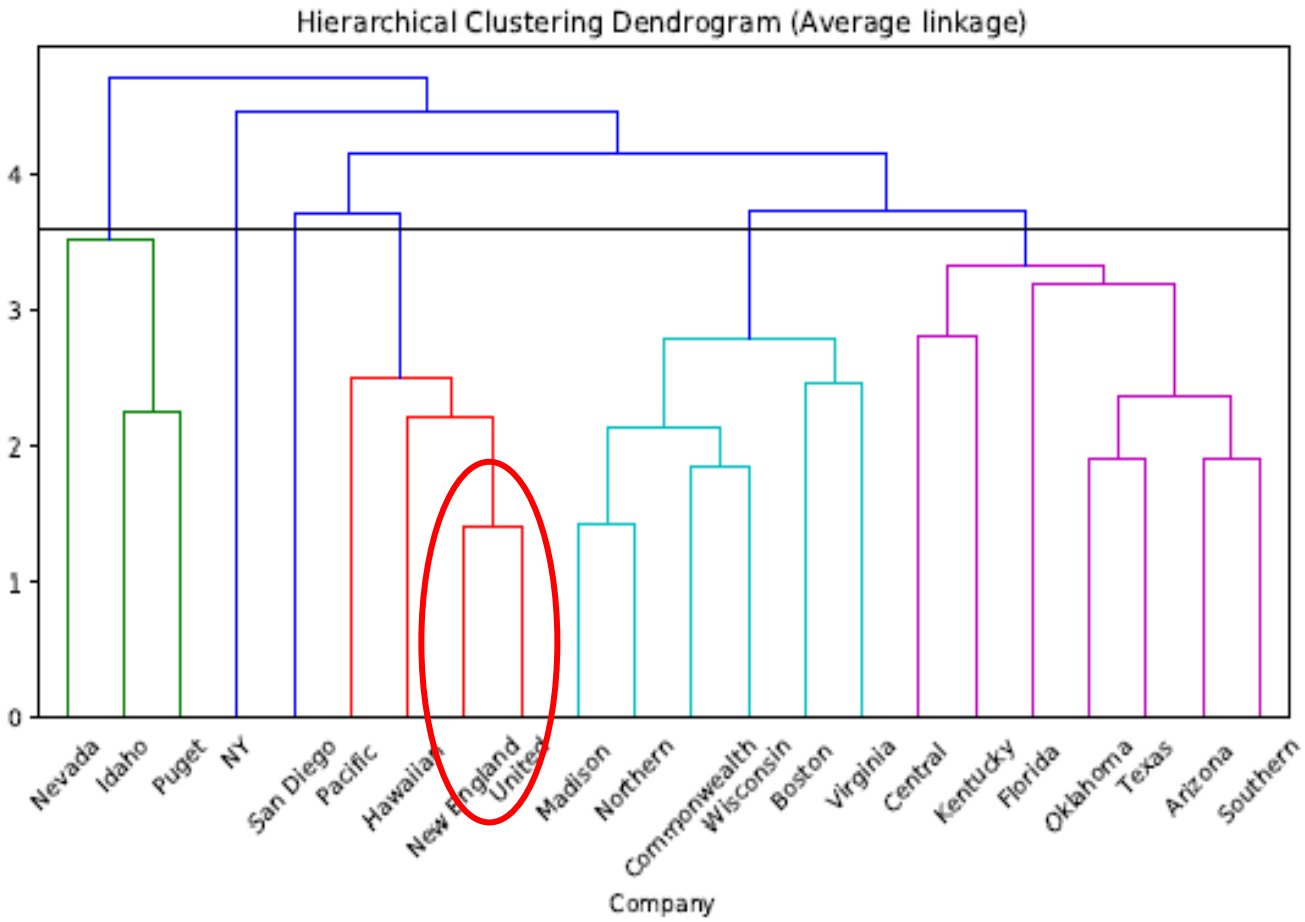
Hierarchical Clustering Steps (Using Agglomerative Method)

1. Start with n clusters (each record is its own cluster)
2. Merge two closest records into one cluster
3. At each successive step, the two clusters closest to each other are merged

* *See illustration with Dendrogram, from bottom up (next slide)*

Hierarchical Clustering

Dendrogram for Average Linkage



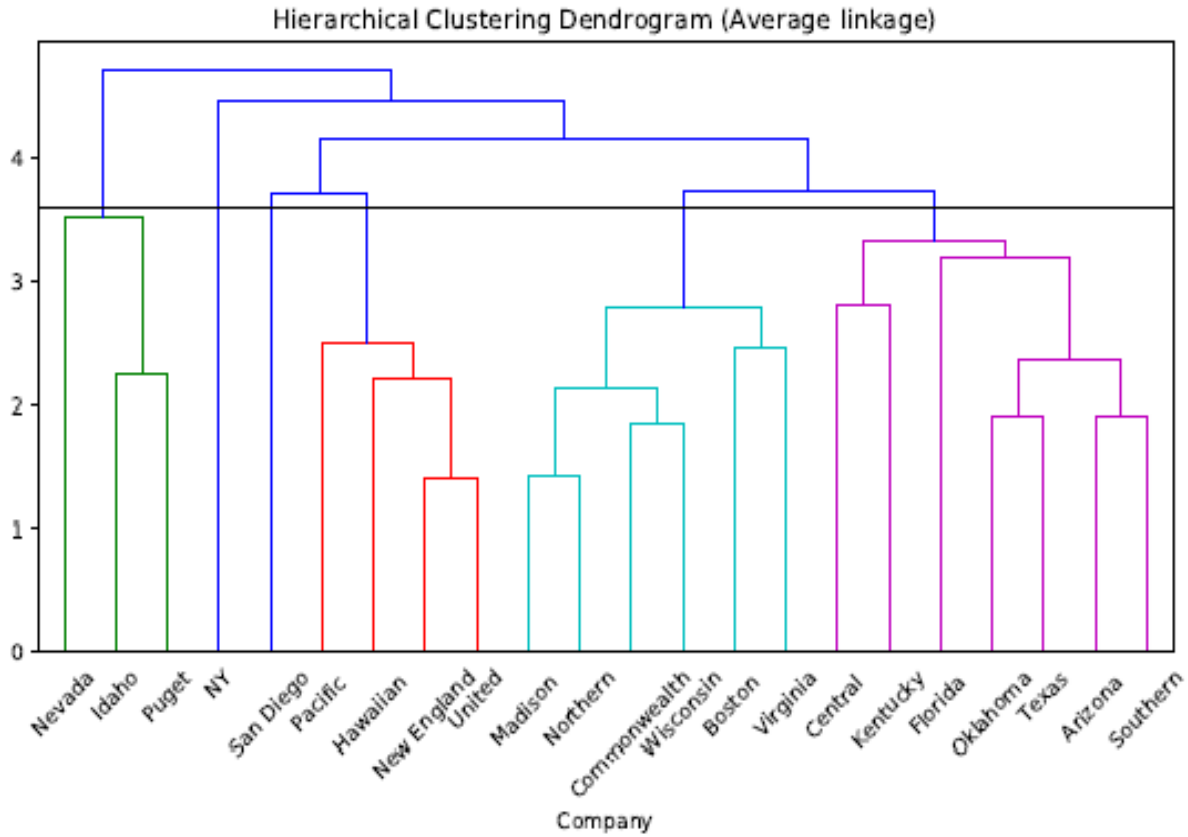
two closest records form first cluster

Reading the Dendrogram

- See process of clustering:
  - Lines connected lower down are merged earlier
  - New England and United will be merged first, then Madison and Northern
- Determining number of clusters:
  - For a given "distance between clusters", a horizontal line intersects the clusters that far apart, to create clusters
  - At distance of 3.6 (horizontal line) data can be reduced to 6 clusters: i.e., 2 singletons, and 4 clusters

Hierarchical Clustering

Dendrogram for Average Linkage



Finding Cluster Membership

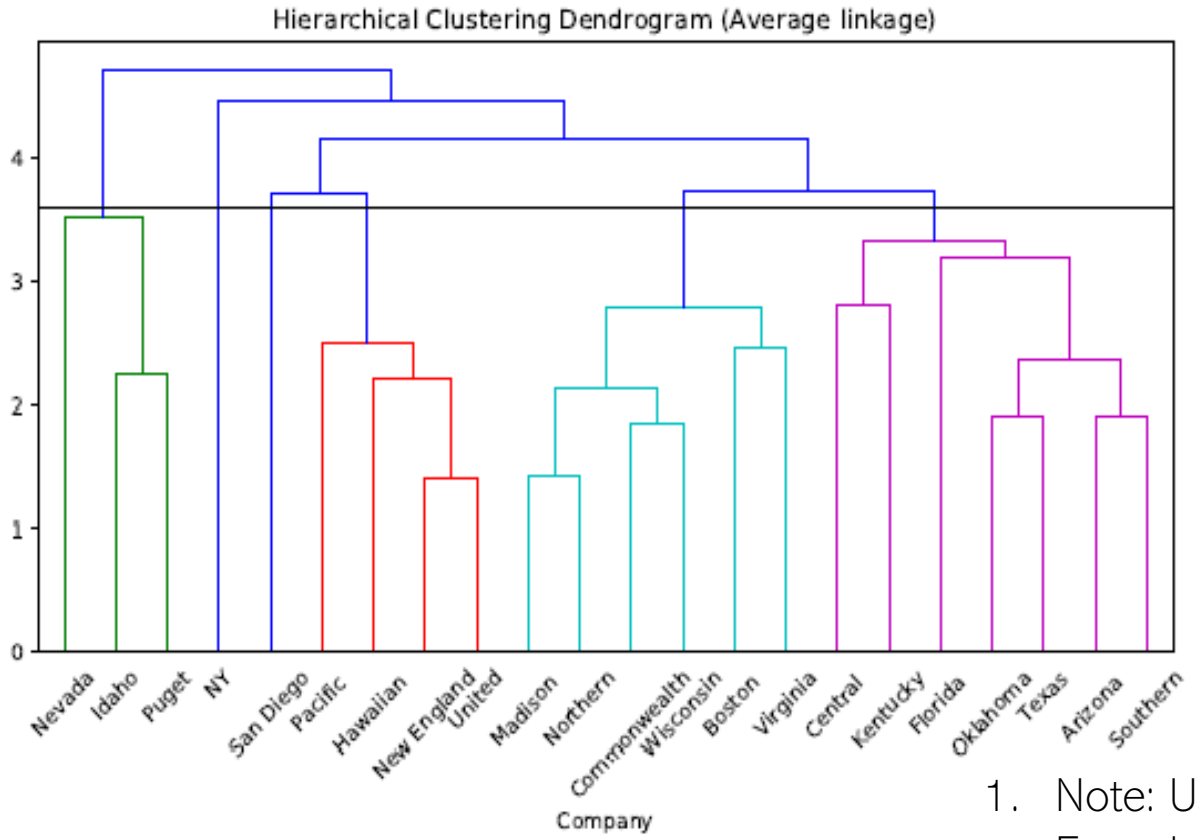
Average Linkage (output modified for clarity)

```
> memb = fcluster(linkage(utilities_df_norm,
   method='average'), 6, criterion='maxclust')
> memb = pd.Series(memb, index=utilities_df_norm.index)
> for key, item in memb.groupby(memb):
> print(key, ': ', ', '.join(item.index))
```

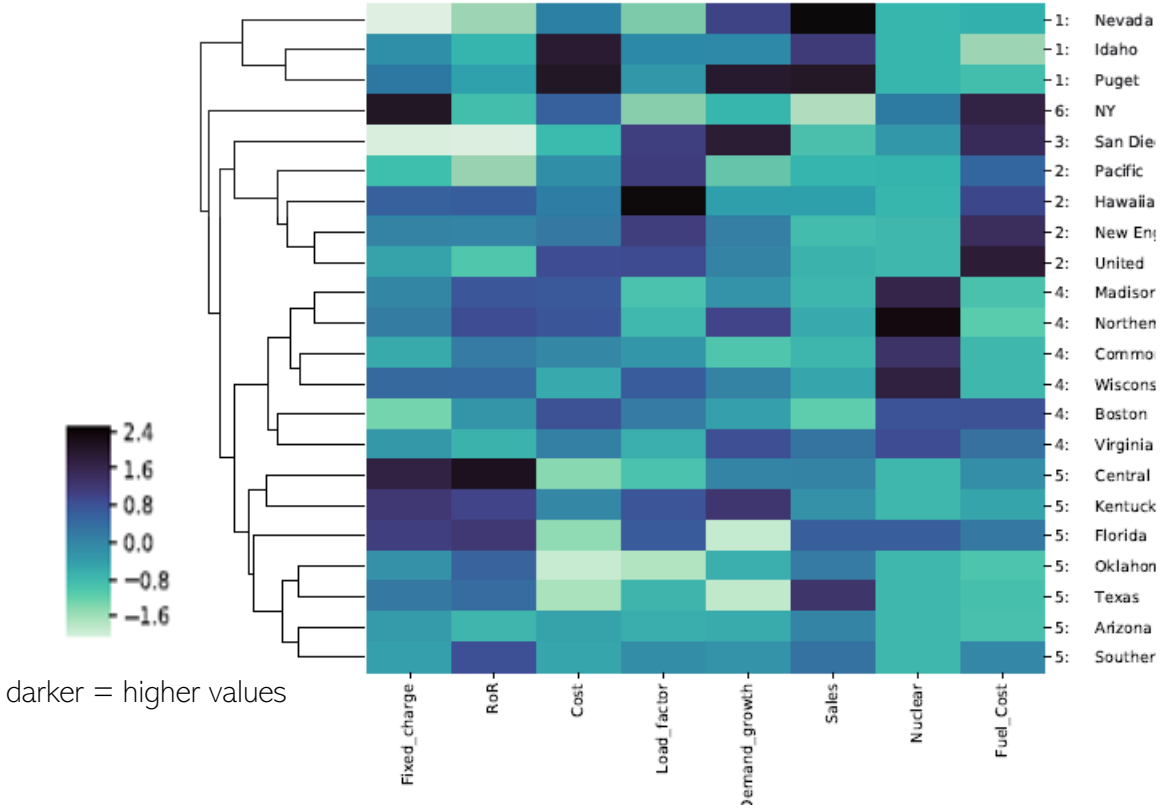
1 : Idaho, Nevada, Puget
2 : Hawaiian, New England, Pacific, United
3 : San Diego
4 : Boston, Commonwealth, Madison, Northern, Wisconsin, Virginia
5 : Arizona, Central, Florida, Kentucky, Oklahoma, Southern, Texas
6 : NY

Hierarchical Clustering

Dendrogram for Average Linkage



Visualize Cluster Features with Heatmap

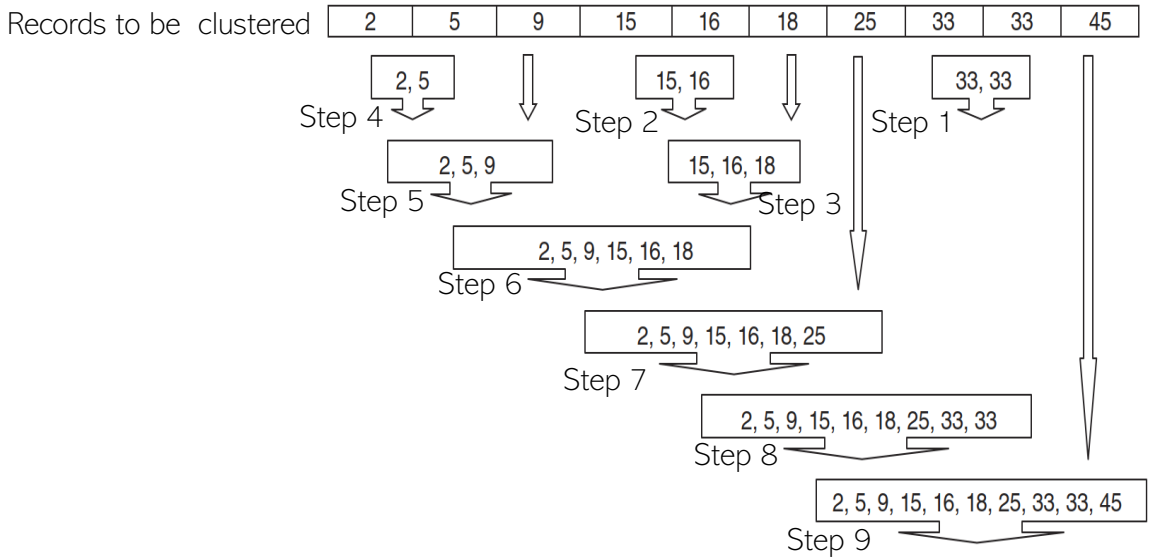


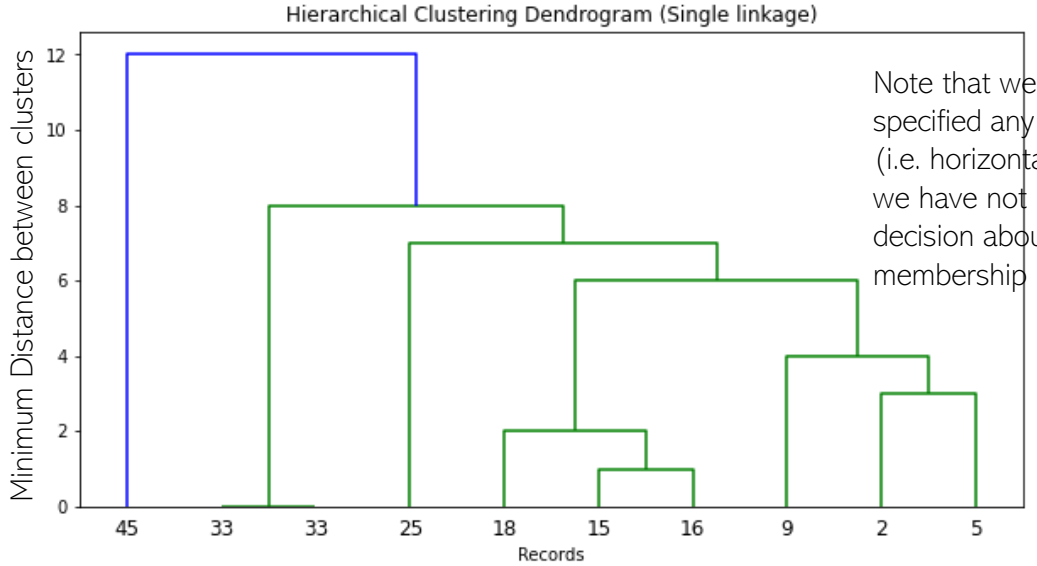
darker = higher values

1. Note: Use heatmap only to profile clusters, not to judge cluster usefulness. E.g., cluster 1 members are low on Nuclear and high Sales, etc.
2. To judge cluster validity / usefulness, use measures such as
   - cophenetic distance (see page 73 of Kassambara 2017) and
   - p-value of hierarchical clusters (see page 156 of Kassambara 2017)

Hierarchical Clustering: More examples & explanations

Single Linkage: Steps and Dendrogram



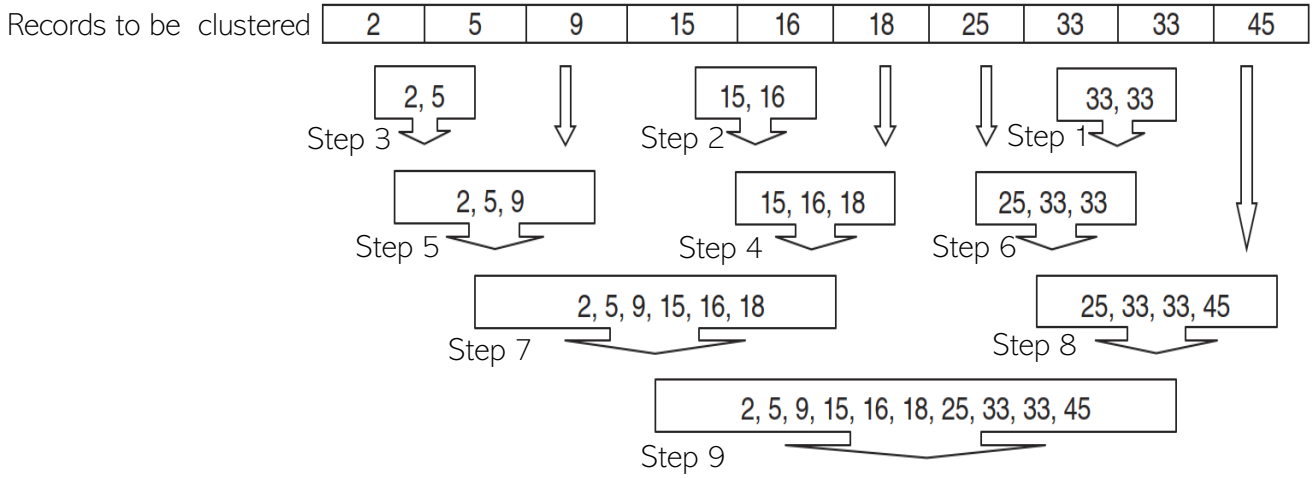


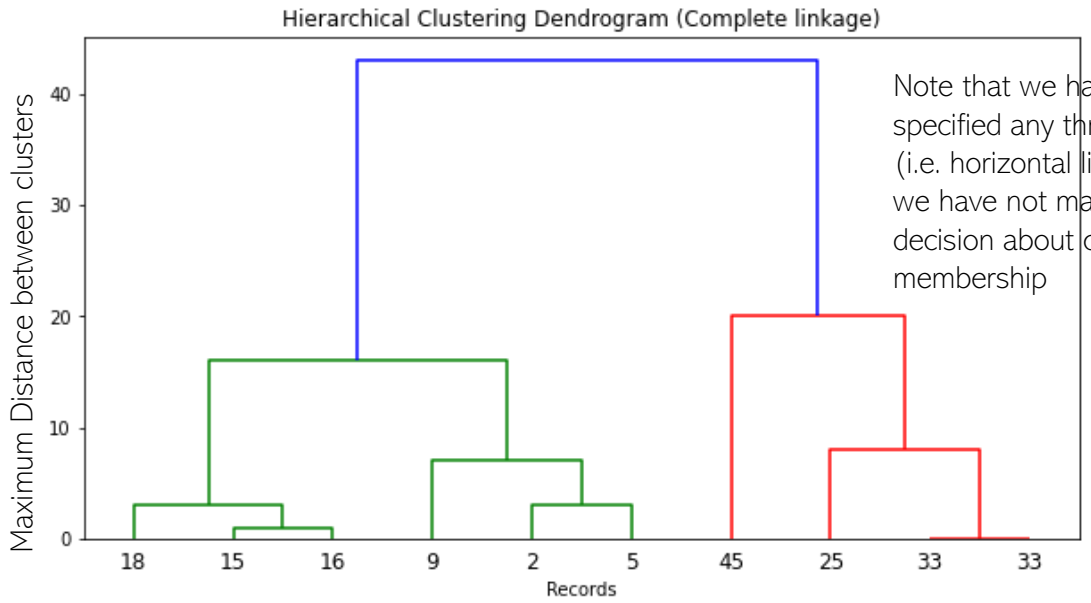
Note that we have not specified any threshold (i.e. horizontal line), so we have not made any decision about cluster membership

- Step 1. The minimum cluster distance is between records with values 33, as the distance between them is 0. Thus, these two clusters are combined into a new cluster of two records {33,33}, Note that after this step, there are (n-1) = 9 clusters remaining.
- Step 2, the clusters containing values 15 and 16 are combined into a new cluster, because their distance of 1 is the minimum between any two clusters remaining.
- Step 3:The cluster containing values 15 and 16 (cluster {15,16}) is combined with cluster {18}, because the distance between 16 and 18 (the closest records in each cluster) is 2, the minimum among remaining clusters.
- Step 4: Clusters {2} and {5} are combined.
- Step 5: Cluster {2,5} is combined with cluster {9}, because the distance between 5 and 9 (the closest records in each cluster) is 4, the minimum among remaining clusters.
- Step 6: Cluster {2,5,9} is combined with cluster {15,16,18}, because the distance between 9 and 15 is 6, the minimum among remaining clusters.
- Step 7: Cluster {2,5,9,15,16,18} is combined with cluster {25}, because the distance between 18 and 25 is 7, the minimum among remaining clusters.
- Step 8: Cluster {2,5,9,15,16,18,25} is combined with cluster {33,33}, because the distance between 25 and 33 is 8, the minimum among remaining clusters.
- Step 9: Cluster {2,5,9,15,16,18,25,33,33} is combined with cluster {45}. This last cluster now contains all the records in the data set.

Hierarchical Clustering: More examples & explanations

Complete Linkage: Steps and Dendrogram





Note that we have not specified any threshold (i.e. horizontal line), so we have not made any decision about cluster membership

- Step 1: As each cluster contains a single record only, there is no difference between single linkage and complete linkage at step 1. The two clusters each containing 33 are again combined.
- Step 2: Just as for single linkage, the clusters containing values 15 and 16 are combined into a new cluster. Again, this is because there is no difference in the two criteria for single-record clusters.
- Step 3: At this point, complete linkage begins to diverge from its predecessor. In single linkage, cluster {15,16} was at this point combined with cluster{18}. But complete linkage looks at the farthest neighbors, not the nearest neighbors. The farthest neighbors for these two clusters are 15 and 18, for a distance of 3. This is the same distance separating clusters {2} and {5}. When there is a tie, we arbitrarily select the first such combination found, therefore combining the clusters{2} and {5} into anew cluster.
- Step 4: Now cluster {15,16} is combined with cluster {18}.
- Step 5: Cluster {2,5} is combined with cluster {9}, because the complete-linkage distance is 7, the smallest among remaining clusters.
- Step 6: Cluster {25} is combined with cluster {33,33}, with a complete-linkage distance of 8.
- Step 7: Cluster {2,5,9} is combined with cluster {15,16,18}, with a complete-linkage distance of 16.
- Step 8: Cluster {25,33,33} is combined with cluster {45}, with a complete-linkage distance of 20.
- Step 9: Cluster {2,5,9,15,16,18} is combined with cluster {25,33,33,45}. All records are now contained in this last large cluster.

Nonhierarchical Clustering: K-Means Clustering

Non-hierarchical Clustering:

K-Means Clustering Algorithm

1. Choose # of clusters desired, *k*
2. Start with a partition into k clusters
   - Often based on random selection of *k* centroids
3. At each step, move each record to cluster with closest centroid
4. Recompute centroids, repeat step 3
5. Stop when moving records increases within-cluster dispersion

Choosing k and Initial Partitioning

- Choose *k* based on how the results will be used
  - e.g., "How many market segments do we want?"
- Also experiment with slightly different *k*'s
- Initial partition into clusters can be random, or based on domain knowledge
  - If random partition, repeat the process with different random partitions

Non-hierarchical Clustering:

K-Means Cluster Membership

```
# Cluster membership
memb = pd.Series(kmeans.labels_, index=utilities_df_norm.index)
for key, item in memb.groupby(memb):
print(key, ': ', ', '.join(item.index))
```

Output

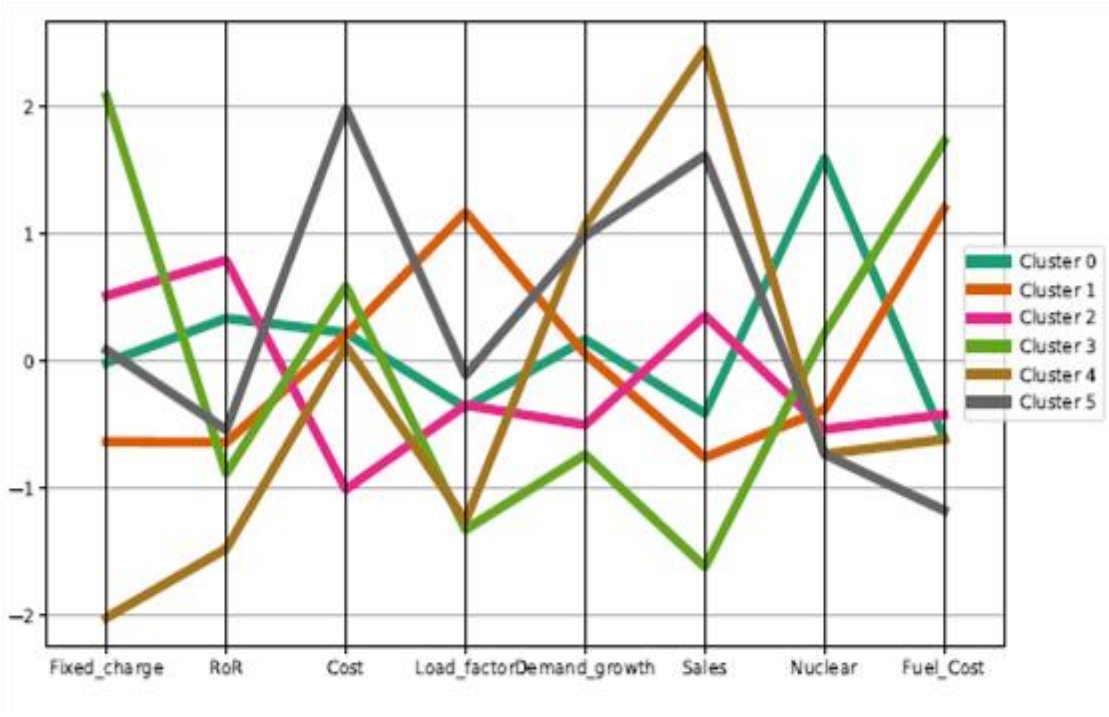
0 : Commonwealth, Madison , Northern, Wisconsin, Virginia
1 : Boston , Hawaiian , New England, Pacific , San Diego, United
2 : Arizona , Central , Florida , Kentucky, Oklahoma, Southern, Texas
3 : NY
4 : Nevada
5 : Idaho, Puget

K-Means Cluster Centroids

```
> centroids = pd.DataFrame(kmeans.cluster_centers_,
   columns=utilities_df_norm.columns)
> pd.set_option('precision', 3)
> centroids
```

	Fixed_ch.	RoR	Cost	Load_f.	Demand_g.	Sales	Nuclear	Fuel_Cost
0	-0.012	0.339	0.224	-0.366	0.170	-0.411	1.602	-0.609
1	-0.633	-0.640	0.207	1.175	0.058	-0.758	-0.381	1.204
2	0.516	0.798	-1.009	-0.345	-0.501	0.360	-0.536	-0.420
3	2.085	-0.883	0.592	-1.325	-0.736	-1.619	0.219	1.732
4	-2.020	-1.476	0.120	-1.257	1.070	2.458	-0.731	-0.616
5	0.088	-0.541	1.996	-0.110	0.988	1.621	-0.731	-1.175

Profile Plot: K-Means Clustering



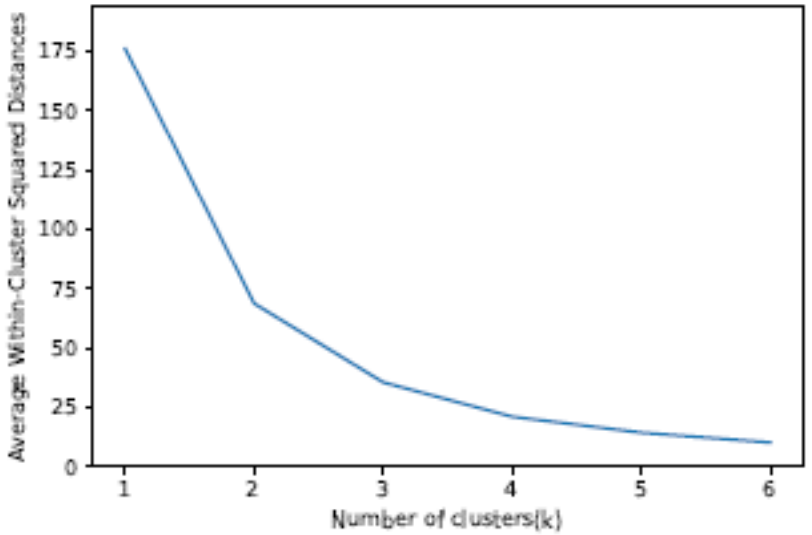
Non-hierarchical Clustering:

K-Means: How Tight are the Clusters?
(Within Cluster Distances)

```
# distances of each record to the cluster centers
distances = kmeans.transform(utilities_df_norm)
# find closest cluster for each record
minSquaredDistances = distances.min(axis=1) ** 2
# combine with cluster labels into a data frame
df = pd.DataFrame({'squaredDistance': minSquaredDistances,
   'cluster': kmeans.labels_}, index=utilities_df_norm.index)
# group by cluster and print information
for cluster, data in df.groupby('cluster'):
   count = len(data)
   withinClustSS = data.squaredDistance.sum()
   print(f'Cluster {cluster} ({count} members):
      {withinClustSS:.2f} within cluster ')
```

```
Output
Cluster 0 (5 members): 10.66 within cluster
Cluster 1 (6 members): 22.20 within cluster
Cluster 2 (7 members): 27.77 within cluster
Cluster 3 (1 members): 0.00 within cluster
Cluster 4 (1 members): 0.00 within cluster
Cluster 5 (2 members): 2.54 within cluster
```

Elbow Plot: Different Choices for *k*



As the number of clusters increases, the cluster members get closer to each other.

Validating Clusters

Validating Clusters

Interpreting Clusters

- **Goal:**
  obtain meaningful and useful clusters
- **Caveats:**
  - Random chance can often produce apparent clusters
  - Different cluster methods produce different results
- **Solutions:**
  - Obtain summary statistics
  - Review clusters in terms of variables **not** used in clustering
  - Label the cluster (e.g., clustering of financial firms in 2008 might yield label like "midsize, sub-prime loser")

Desirable Cluster Features

- **Stability**
  - are clusters and cluster assignments sensitive to slight changes in inputs?
  - Are cluster assignments in partition B similar to partition A?
- **Separation**
  - check ratio of between-cluster variation to within-cluster variation (higher is better)
- *See Kassambara, A. (2017) (uploaded to Canvas) for several techniques and metrics for cluster validation in R.*

Clustering: Summary

- Cluster analysis is an exploratory tool
  - Useful only when it produces **meaningful** clusters
- **Hierarchical** clustering gives visual representation of different levels of clustering
  - On other hand, due to non-iterative nature, it can be unstable, can vary highly depending on settings, and is computationally expensive
- **Non-hierarchical** is computationally cheap and more stable; requires user to set *k*
- Can use both methods (e.g., for initial idea of *k*)
- Be wary of chance results; data may not have definitive "real" clusters

Class 5 Exercise

(Submit on canvas in today's Class Participation folder. Due before next class):

1. Upload your completed practice python file for Class 6.

2. [1/2 to 1page] In choosing the number of clusters (k) in k-Means clustering, compare the Elbow Method to
    I. The Silhouette Method
    II. The Gap Statistic Method

    Which method, among the three, would you consider as the best? Why?

3. [1/2 page] Describe the DBSCAN (Density-Based Clustering) algorithm. Does it offer any improvements over hierarchical and k-Means algorithms?

Next Class …

- In-person at Room W133 SSB
- Quiz 2 (online at the beginning of class, bring your laptops)
- Text Analytics

Looking forward to meeting you!