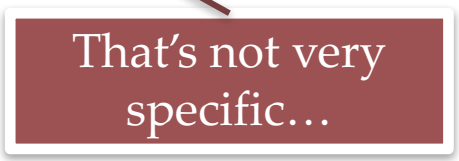
Cluster Analysis

(see also: Segmentation)

Cluster Analysis

- **Unsupervised**: no target variable for training
- Partition the data into groups (clusters) so that:
  - Observations within a cluster are similar in some sense
  - Observations in different clusters are different in some sense
- There is no one correct answer, though there are good and bad clusters
- No method words best all the time



(Some) Applications of Clustering

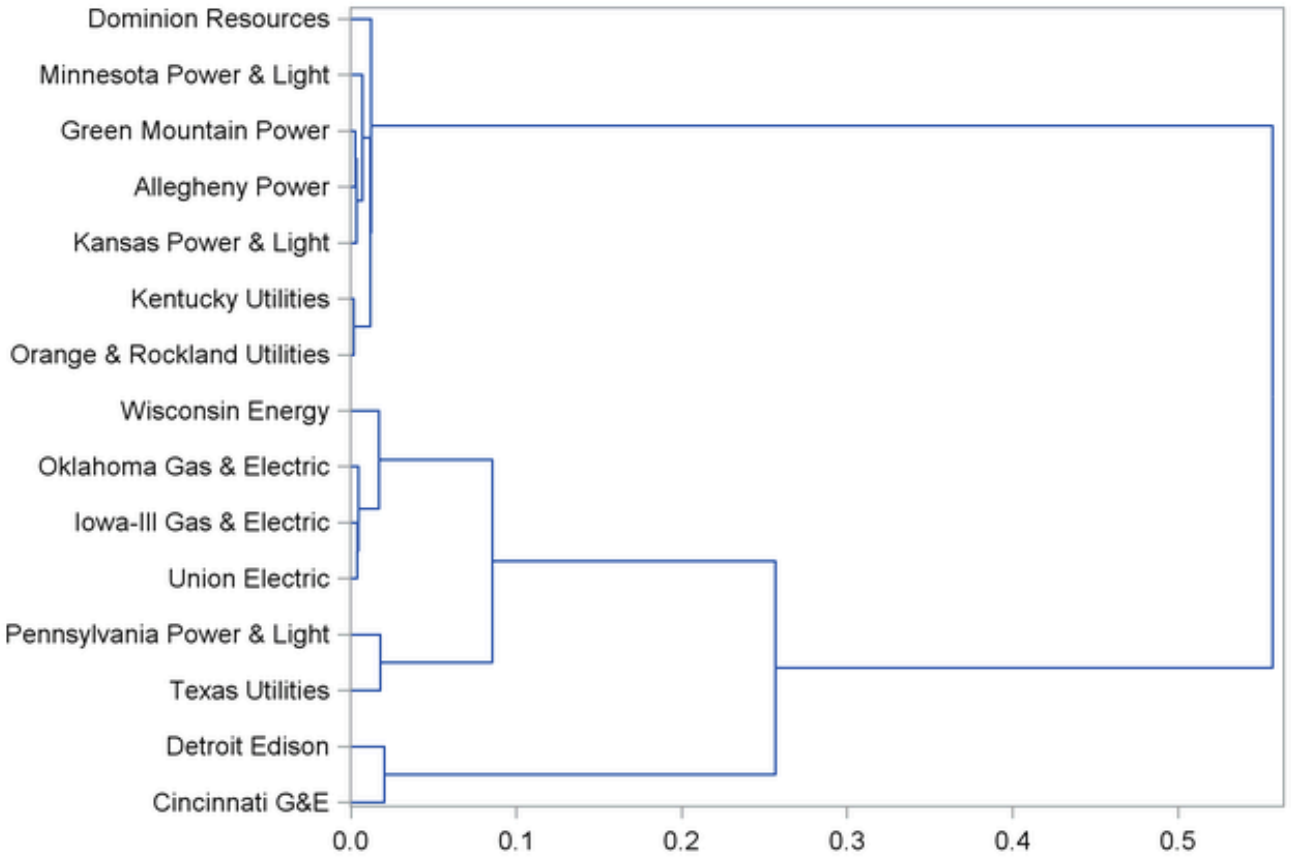
- Customer segmentation: groups of customers with similar shopping or buying patterns
- Dimension reduction:
  - cluster variables together
  - cluster individuals together and use cluster variable as proxy for demographic or behavioral variables
- Image segmentation
- Gather stores with similar characteristics for sales forecasting
- Find related topics in text data
- Find communities in social networks

Methodology

- Hard vs. Fuzzy Clustering
  - **Hard:** objects can belong to only one cluster
    - k-means (PROC FASTCLUS)
    - DBSCAN
    - Hierarchical (PROC CLUSTER)
  - **Fuzzy:** objects can belong to more than one cluster (usually with some probability)
    - Gaussian Mixture Models

Methodology

- Hierarchical vs. Flat
  - **Hierarchical:** clusters form a tree so you can visually see which clusters are most similar to each other.



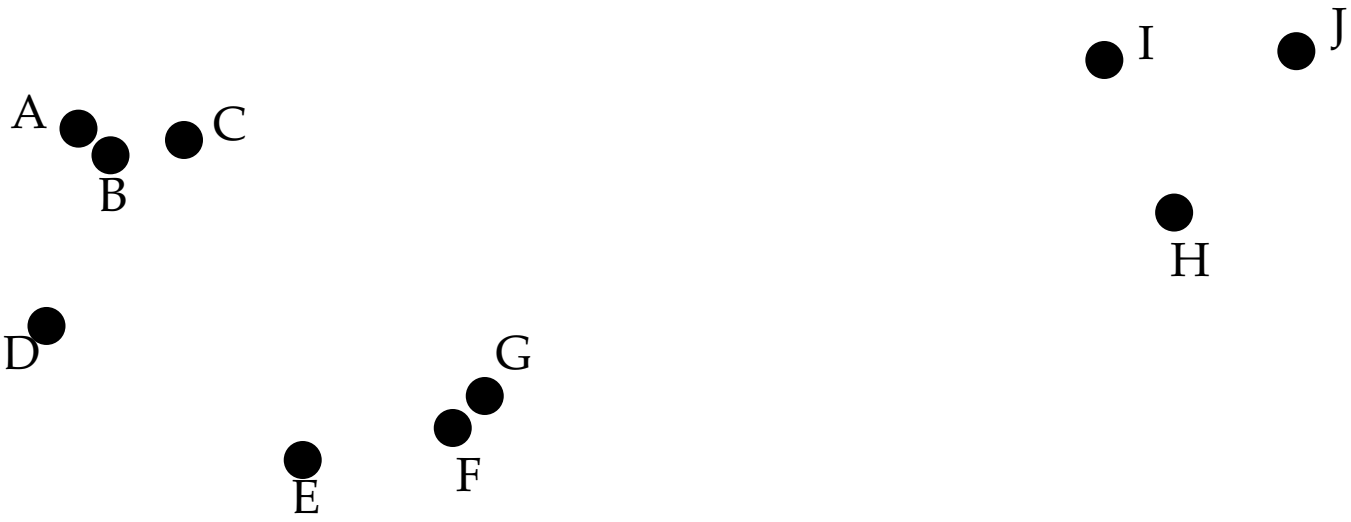
Methodology

- Hierarchical vs. Flat
  - **Hierarchical:** clusters form a tree so you can visually see which clusters are most similar to each other.
    - **Agglomerative:** points start out as individual clusters, and they are combined until everything is in one cluster.
    - **Divisive:** All points start in same cluster and at each step a cluster is divided into two clusters.
  - **Flat:** Clusters are created according to some other process, usually iteratively updating cluster assignments

Hierarchical Clustering

(Agglomerative)

Some Data



Hierarchical Clustering

(Agglomerative)

First Step

Hierarchical Clustering

(Agglomerative)

Second Step

Hierarchical Clustering

(Agglomerative)

Third Step

Hierarchical Clustering

(Agglomerative)

Forth Step

Hierarchical Clustering

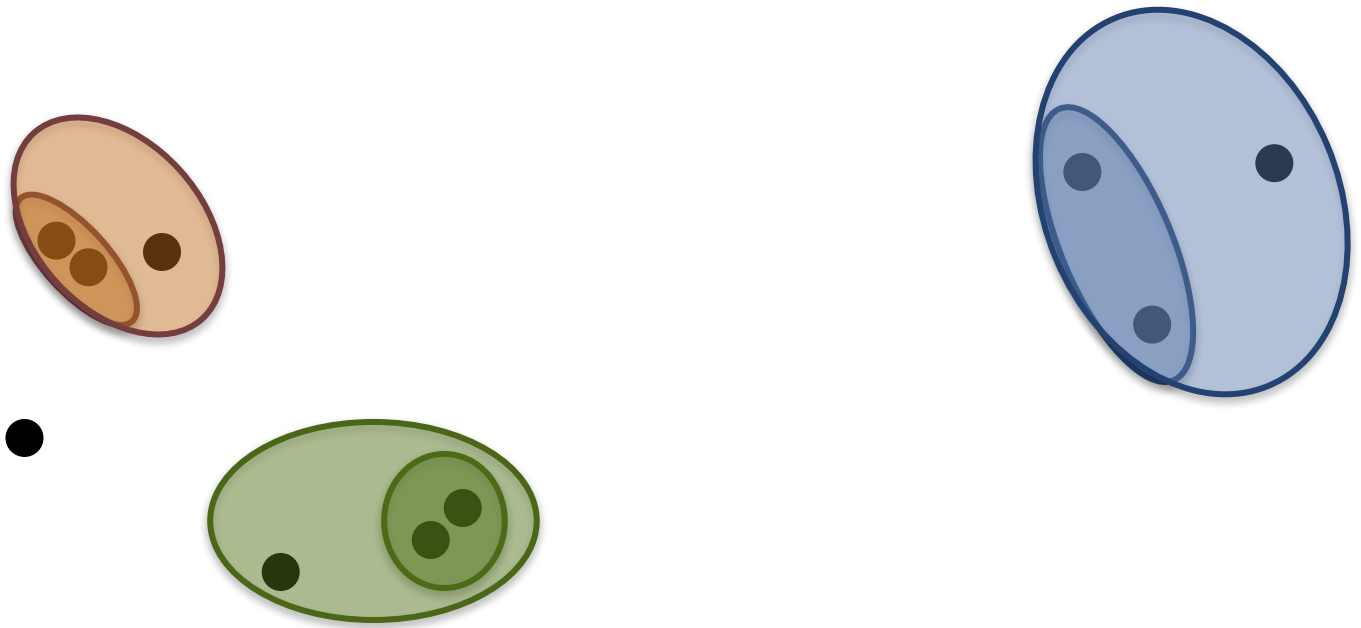
(Agglomerative)

Fifth Step

Hierarchical Clustering

(Agglomerative)

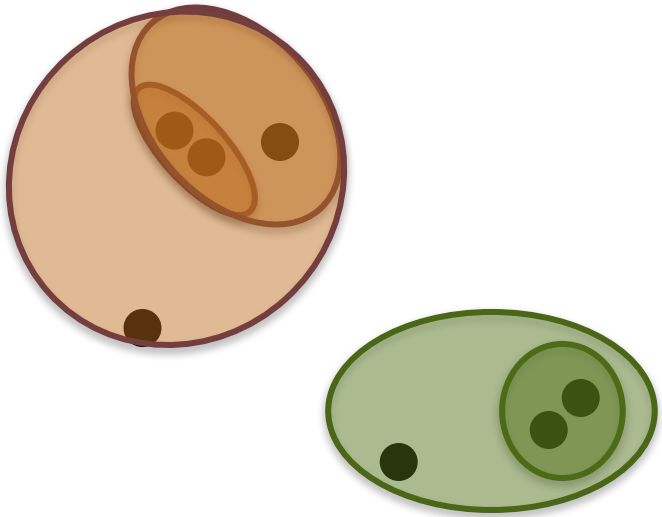
Sixth Step



Hierarchical Clustering

(Agglomerative)

Seventh Step

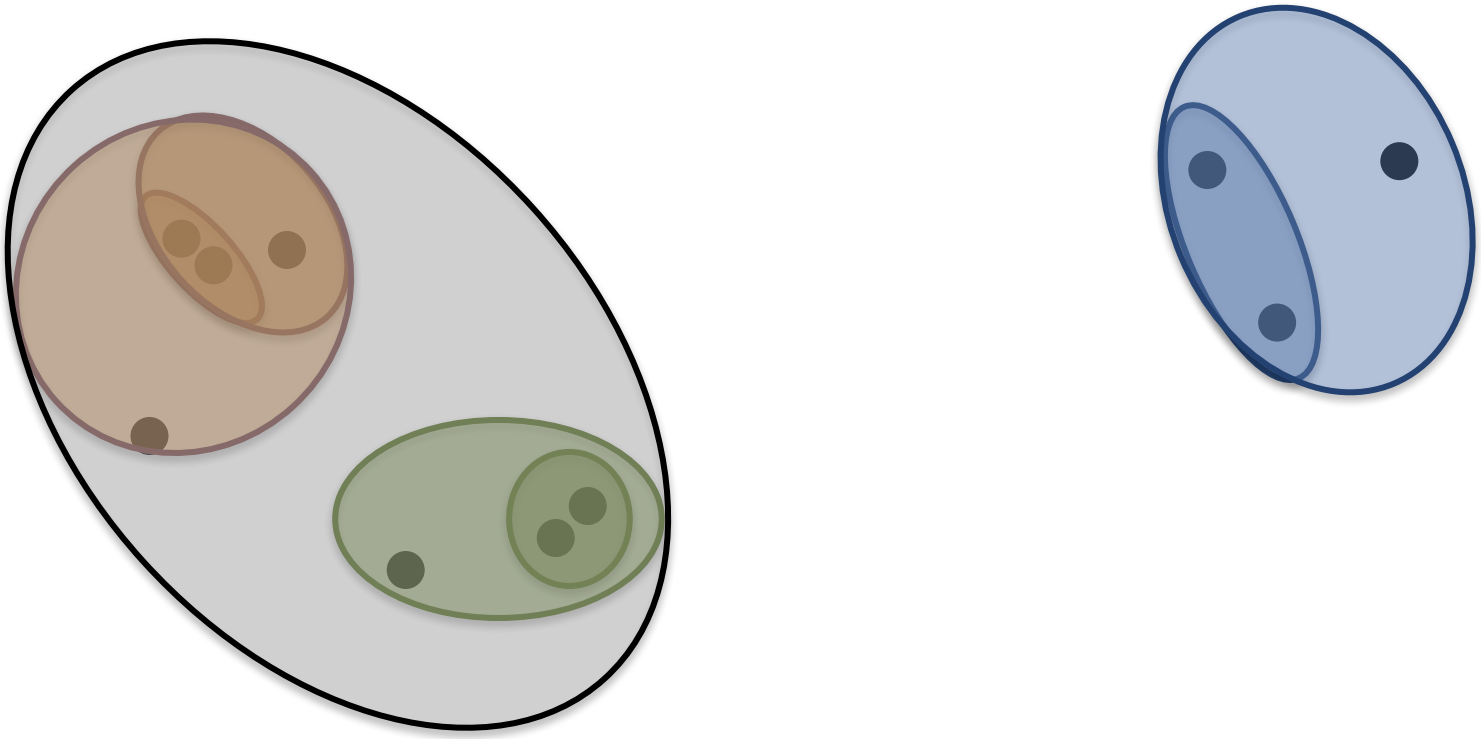


We might have known that we only wanted 3 clusters, in which case we'd stop once we had 3.

Hierarchical Clustering

(Agglomerative)

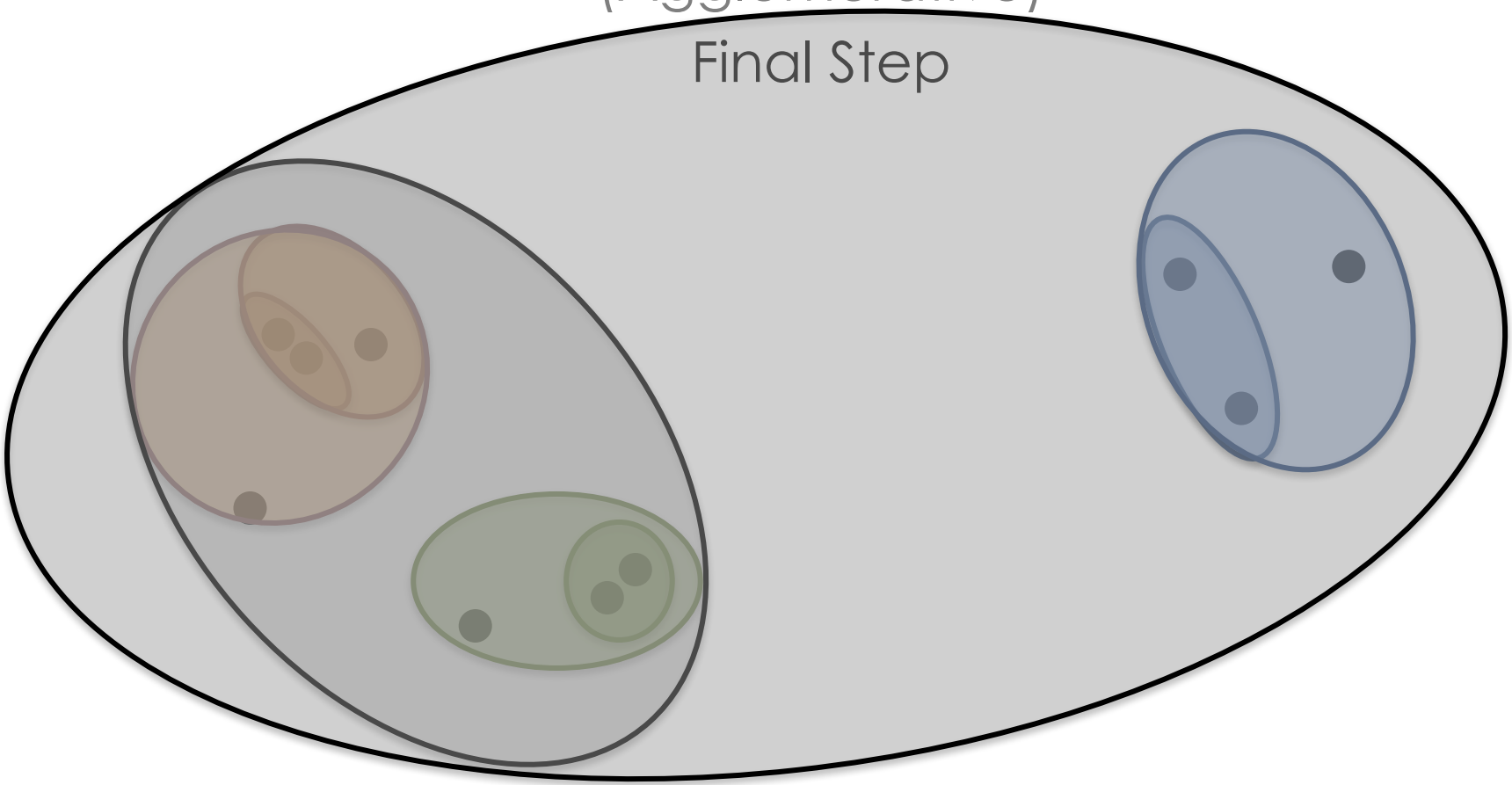
Eighth Step



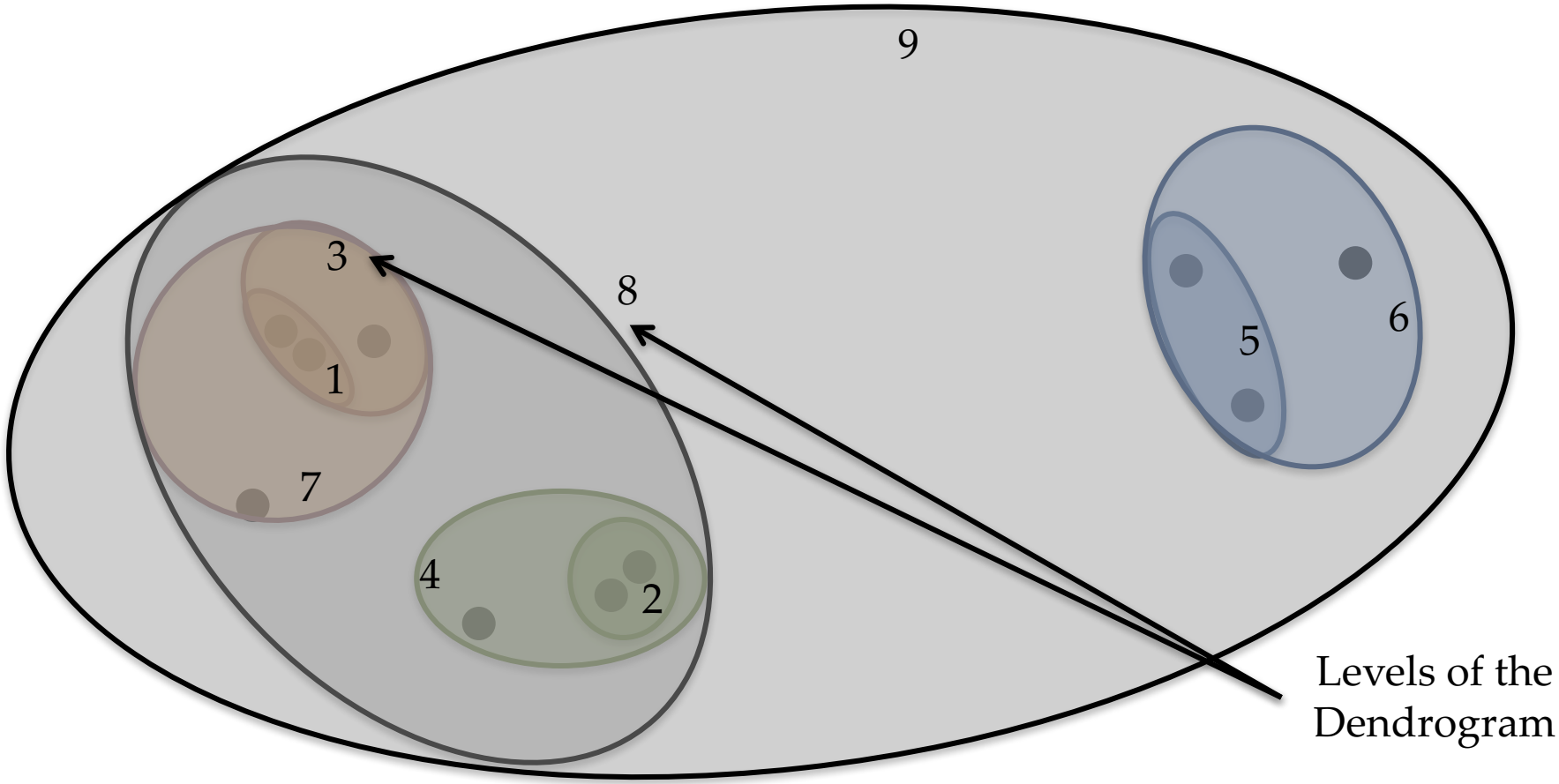
Hierarchical Clustering

(Agglomerative)

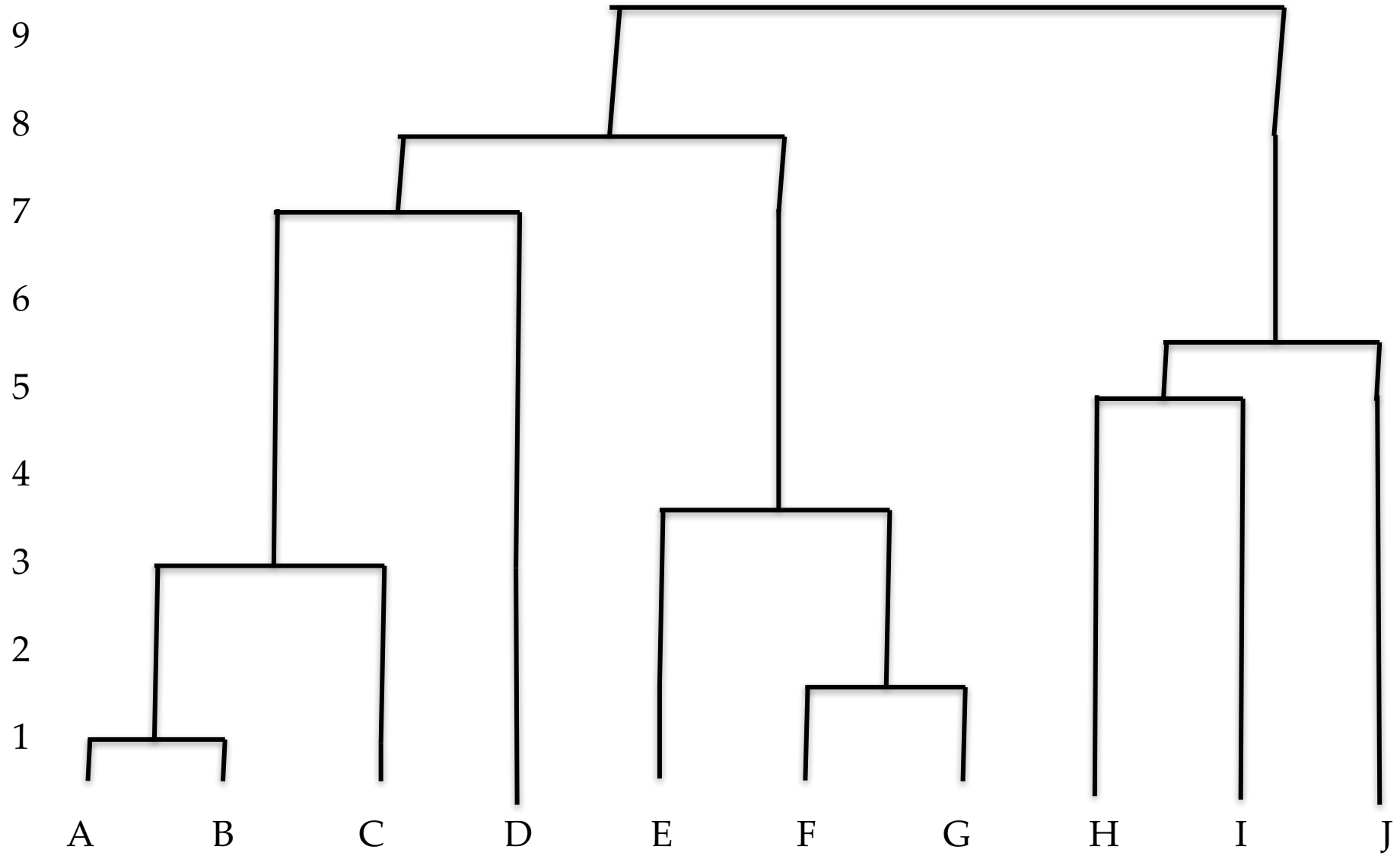
Final Step



Hierarchical Clustering



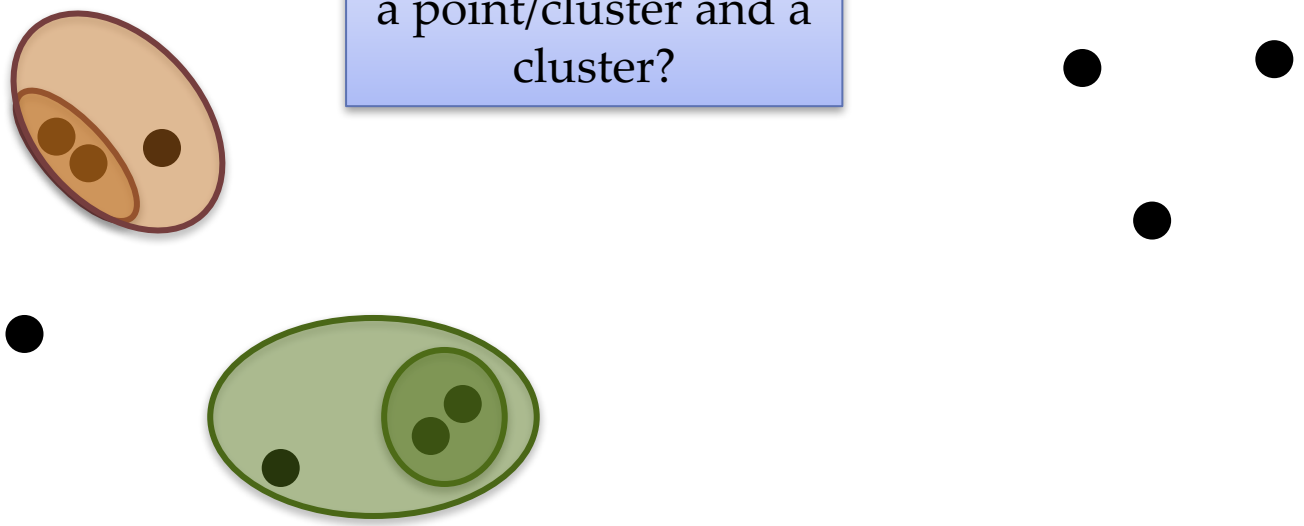
Resulting Dendrogram



Linkages

Which clusters/points are closest to each other?

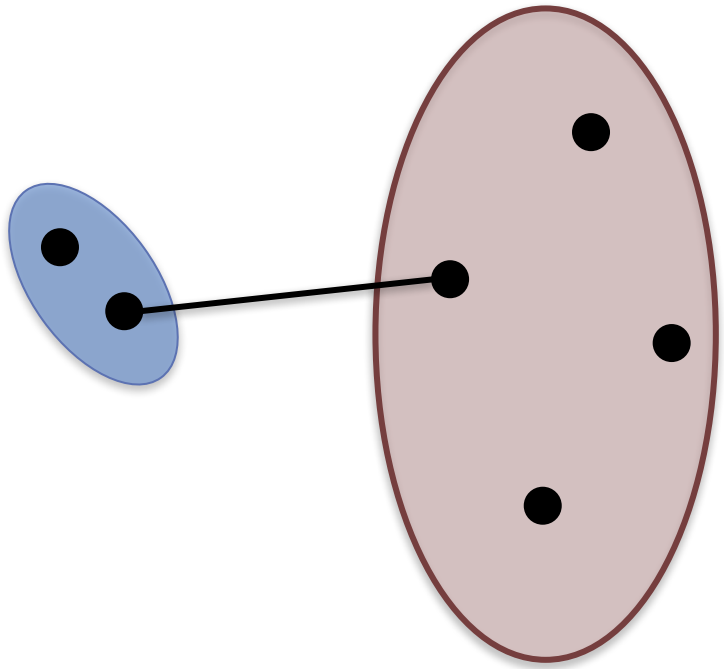
How do I measure the distance between a point/cluster and a cluster?



Linkages

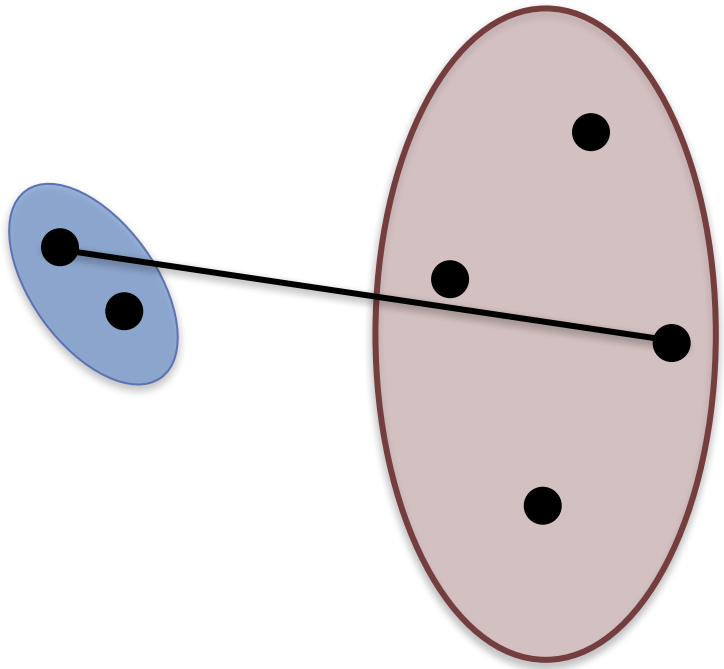
Single Linkage: Distance between the closest points in the clusters.

(Minimum Spanning Tree)



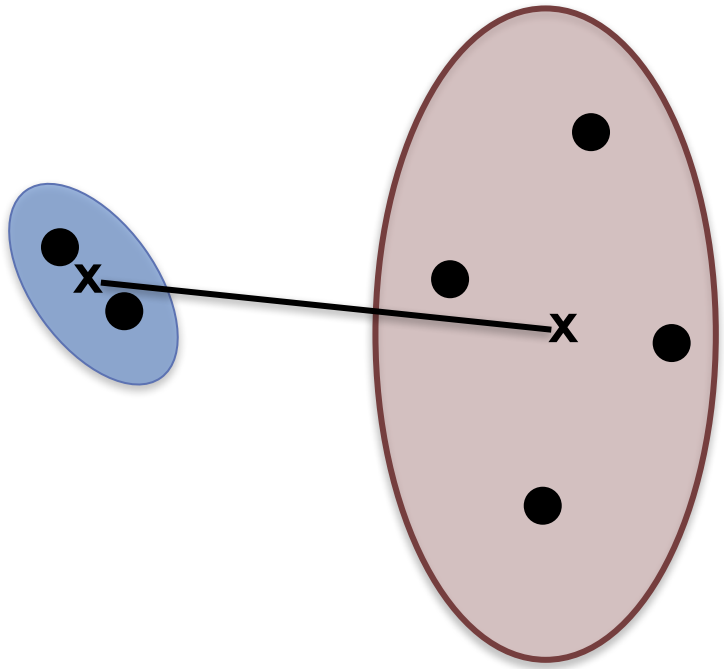
Linkages

Complete Linkage: Distance between the farthest points in the clusters.



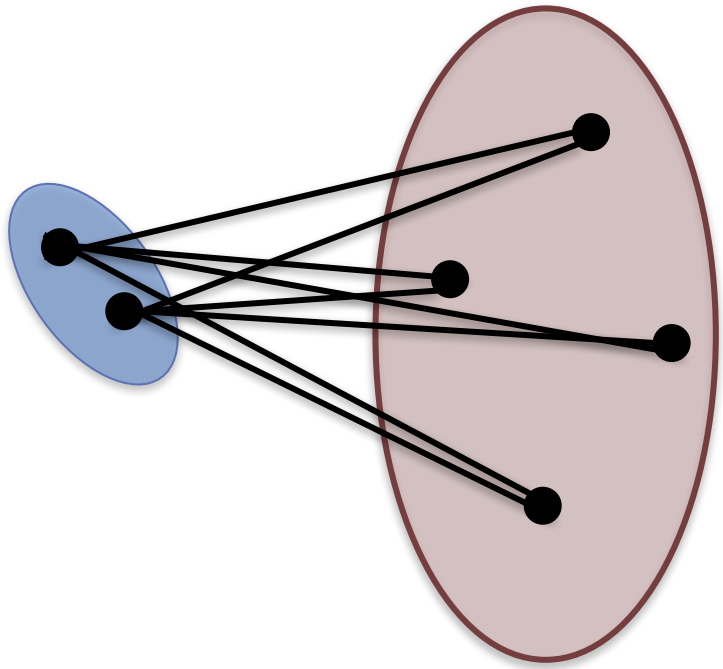
Linkages

Centroid Linkage: Distance between the centroids (means) of each cluster.



Linkages

Average Linkage: Average distance between all points in the clusters.

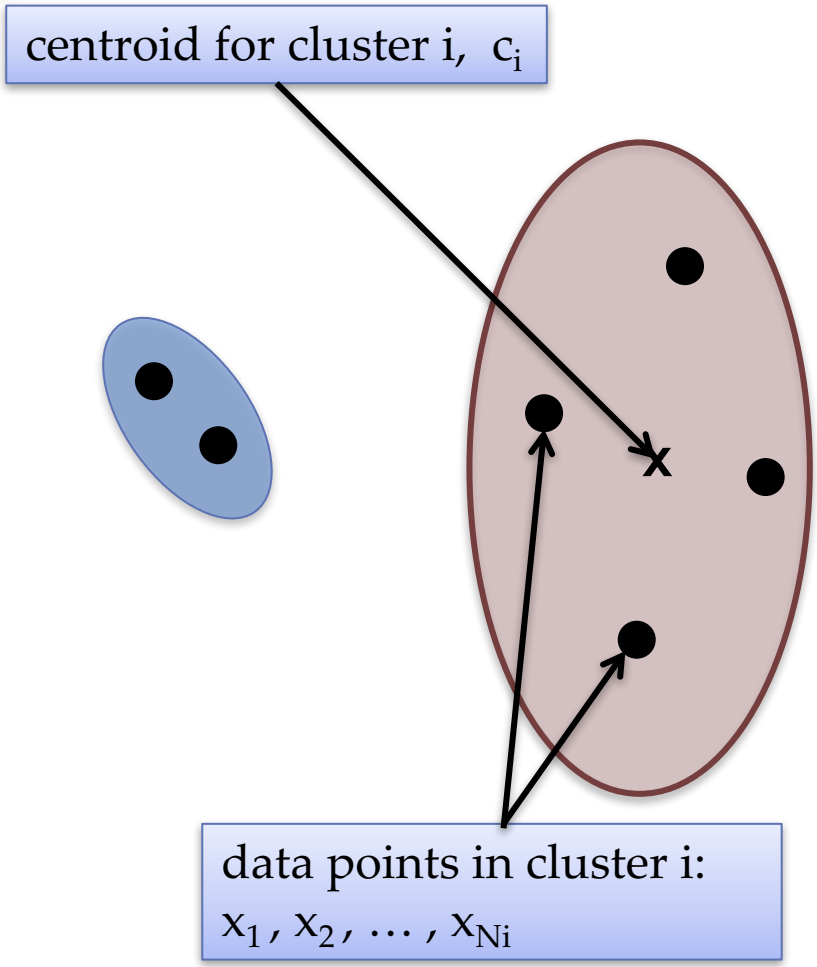


Linkages

Ward's Method: Increase in SSE (variance) when clusters are combined.

$$\sum_{j=1}^{N_i} \|\mathbf{x}_j - \mathbf{c}_i\|_2$$

- Default in SAS **PROC CLUSTER**
- Shown mathematically similar to centroid linkage



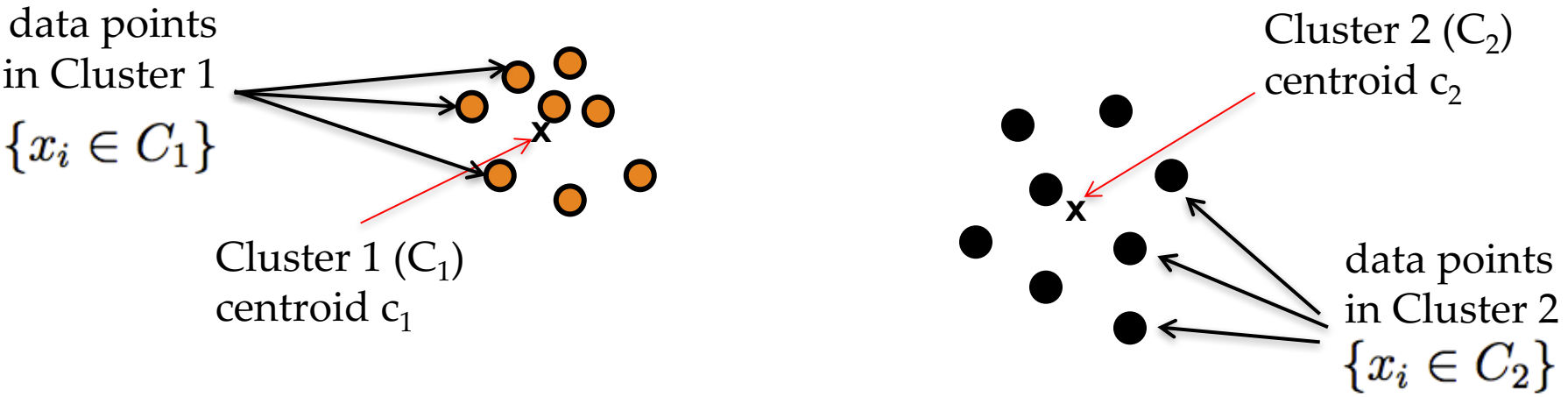
Hierarchical Clustering Summary

- Disadvantages
  - Lacks global objective function: only makes decision based on local criteria.
  - Merging decisions are final. Once a point is assigned to a cluster, it stays there.
  - Computationally intensive, large storage requirements, not good for large datasets
  - Poor performance on noisy or high-dimensional data like text.
- Advantages
  - Lacks global objective function: no complicated algorithm or problem with local minima
  - Creates hierarchy that can help choose the number of clusters and examine how those clusters relate to each other.
  - Can be used in conjunction with other faster methods

k-Means Clustering

(PROC FASTCLUS in SAS)

- The most popular clustering algorithm



- Tries to minimize the sum of squared distances from each point to its cluster centroid. (Global objective function)

$$\sum_{C_k} \sum_{x_i \in \mathbf{C}_k} \|x_i - c_k\|^2$$

k-Means Algorithm

- Start with k "seed points"
  - Randomly initialized (most software)
  - Determined 'methodically' (SAS PROC FASTCLUS)
- Assign each data point to the closest seed point.
- The seed point then represents a cluster of data
- Reset seed points to be the centroids of the cluster
- Repeat steps 2-4 updating the cluster centroids until they do not change.

k-Means Interactive Demo

http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

(You may have to add the site to your exceptions list on the Java Control Panel to view.)

Choice of Distance Metric

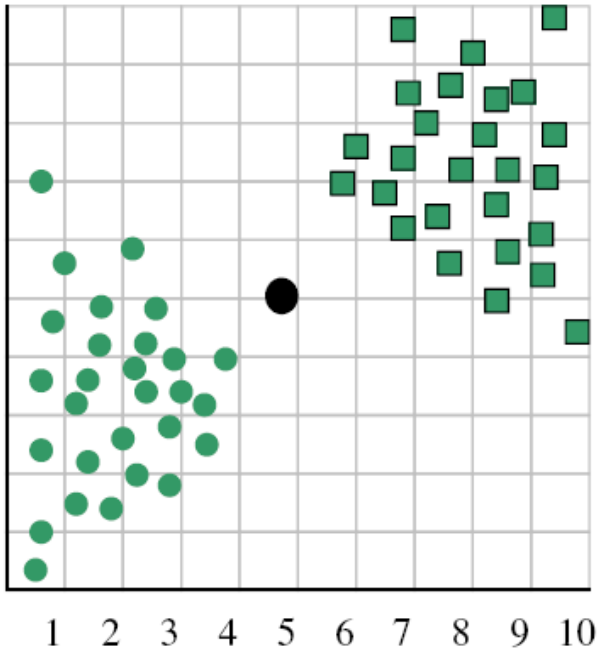
- Most distances like Euclidean, Manhattan, or Max will provide similar answers.

- Use cosine distance (really *1-cos* since cosine measures similarity) for text data. This is called **spherical k-means.**

- Using Mahalanobis distance is essentially the Expectation-Maximization (EM method) for Gaussian Mixtures.

Determining Number of Clusters (SSE)

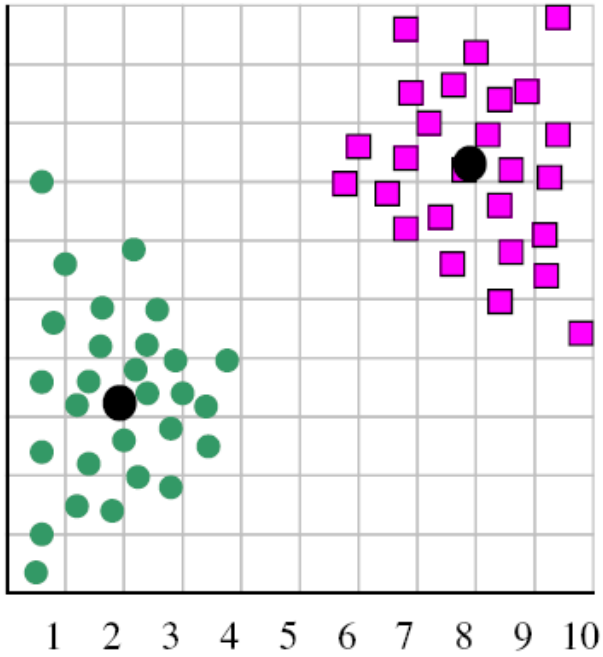
- Try the algorithm with k=1,2,3,…
- Examine the objective function values
- Look for a place where the marginal benefit to objective function for adding a cluster becomes small



k=1 objective function (SSE) is 902

Determining Number of Clusters (SSE)

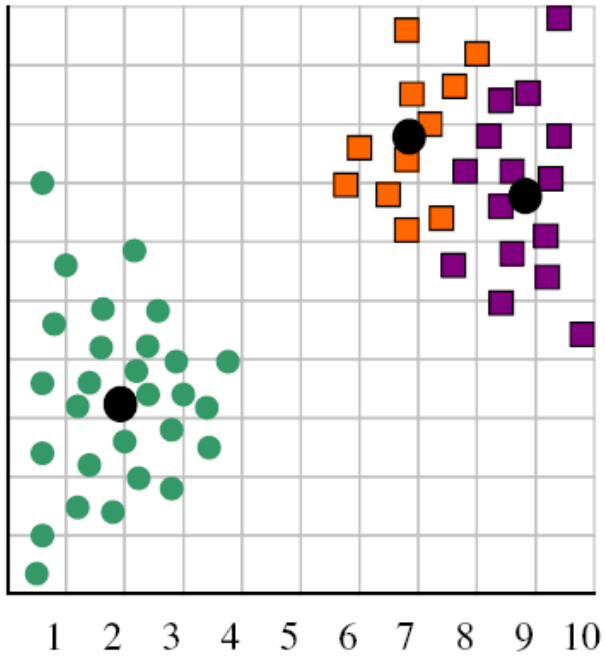
- Try the algorithm with k=1,2,3,…
- Examine the objective function values
- Look for a place where the marginal benefit to objective function for adding a cluster becomes small



k=2 objective function (SSE) is 213

Determining Number of Clusters (SSE)

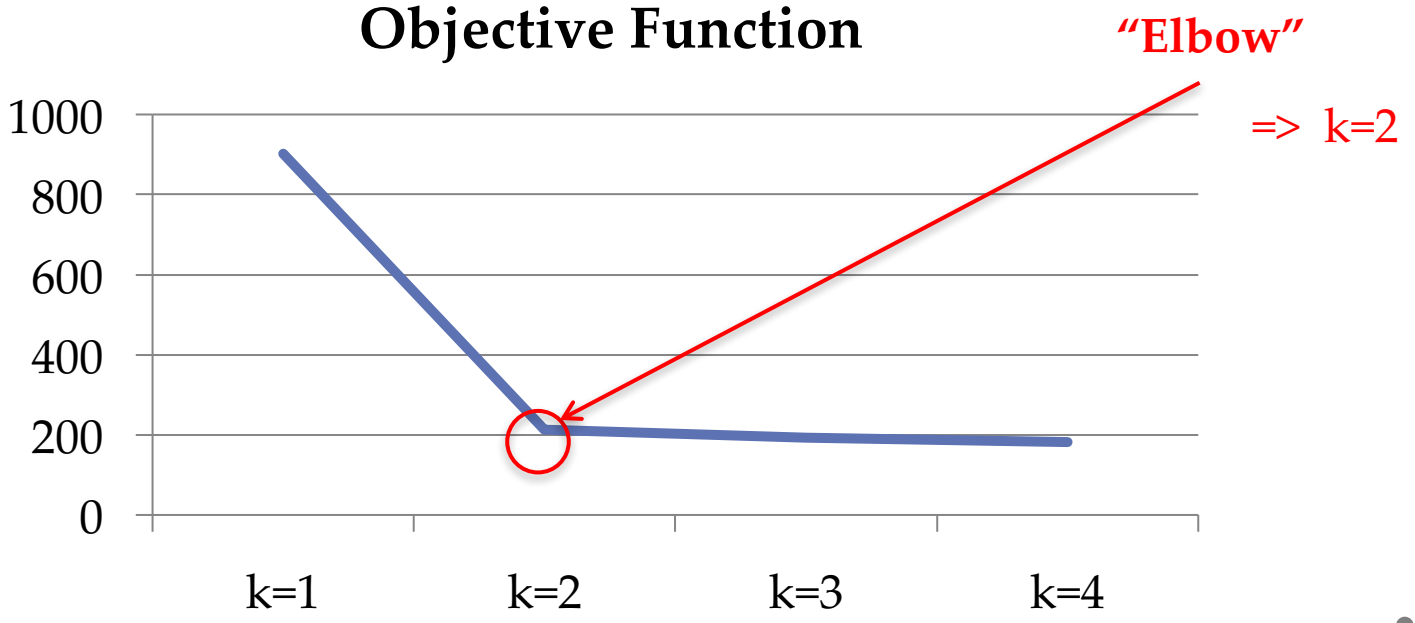
- Try the algorithm with k=1,2,3,…
- Examine the objective function values
- Look for a place where the marginal benefit to objective function for adding a cluster becomes small



k=3 objective function (SSE) is 193

Determining Number of Clusters (SSE)

- Try the algorithm with k=1,2,3,…
- Examine the objective function values
- Look for a place where the marginal benefit to objective function for adding a cluster becomes small



k-Means Summary

- Disadvantages
  - Dependent on initialization (initial seeds)
  - Can be sensitive to outliers
    - If problem, should consider k-mediods (uses median not mean)
  - Have to input the number of clusters
  - Difficulty detecting non-spheroidal (globular) clusters
- Advantages
  - Modest time/storage requirements.
  - Shown you can terminate method after small number of iterations with good results.
  - Good for wide variety of data types

Cluster Validation and Profiling

Additional Notes for Self-Study

Cluster Validation

How do I know that my clusters are actually clusters?

- Lots of techniques/metrics have been proposed
  - Measure **separation** between clusters
  - Measure **cohesion** within clusters
- All have merit, most are difficult to interpret in the context of statistical significance.

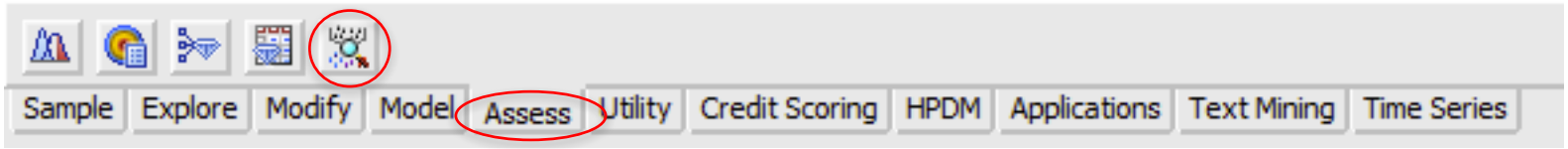
Cluster Validation

- To establish statistical significance:
  - Show that you can't do just as well with randomized data (i.e. assume the null hypothesis of no clusters)
  - Simulate ~1000 random data sets choosing from the distributions or ranges of your variables. Cluster them with the same number of clusters. Record the SSE (k-means objective function) or validity metric of choice. Use this to show that your actual SSE is far better than you could expect to achieve if no clusters exist.

Profiling Clusters

Now that we have clusters, how do we describe them?

- Use basic descriptives and hypothesis tests to show differences between clusters
- Use a decision tree to predict cluster
- SAS EM has segment profiler node



Other types of Clustering (self-study)

- **DBSCAN** – Density based algorithm designed to find dense areas of points. Capable of identifying 'noise' points which do not belong to any clusters.

- **Graph/Network Clustering** – Spectral clustering and modularity maximization. Covered in Social Network Analysis in Spring.

Some Explanation of SAS's Clustering Output (SELF-STUDY)

Because it's not exceedingly easy to figure out online!

Cubic Clustering Criterion (CCC)

- Only available in SAS (to my knowledge)
- CCC > 2 means that clustering is good
- 0 > CCC > 2 means clustering requires examination
- If slightly negative, risk of outliers is low
- If ~< -30 then risk of outliers is high

- Should not be used with single or complete linkage, but with centroid or ward's method.
- Each cluster must have >10 observations.

- Source: Tufféry, Stéphane. *Data Mining and Statistics for Decision Making.* Wiley *2011*

Determining Number of Clusters with the Cubic Clustering Criterion (CCC)

- A partition into k clusters is good when we see a dip in CCC for k-1 clusters and a peak for k clusters.
- After k clusters, the CCC should either a gradually decrease or a gradual rise (the latter event happens when more isolated groups or points are present)[1]

- Source: Tufféry, Stéphane. *Data Mining and Statistics for Decision Making.* Wiley *2011*

Determining Number of Clusters with the Cubic Clustering Criterion (CCC)

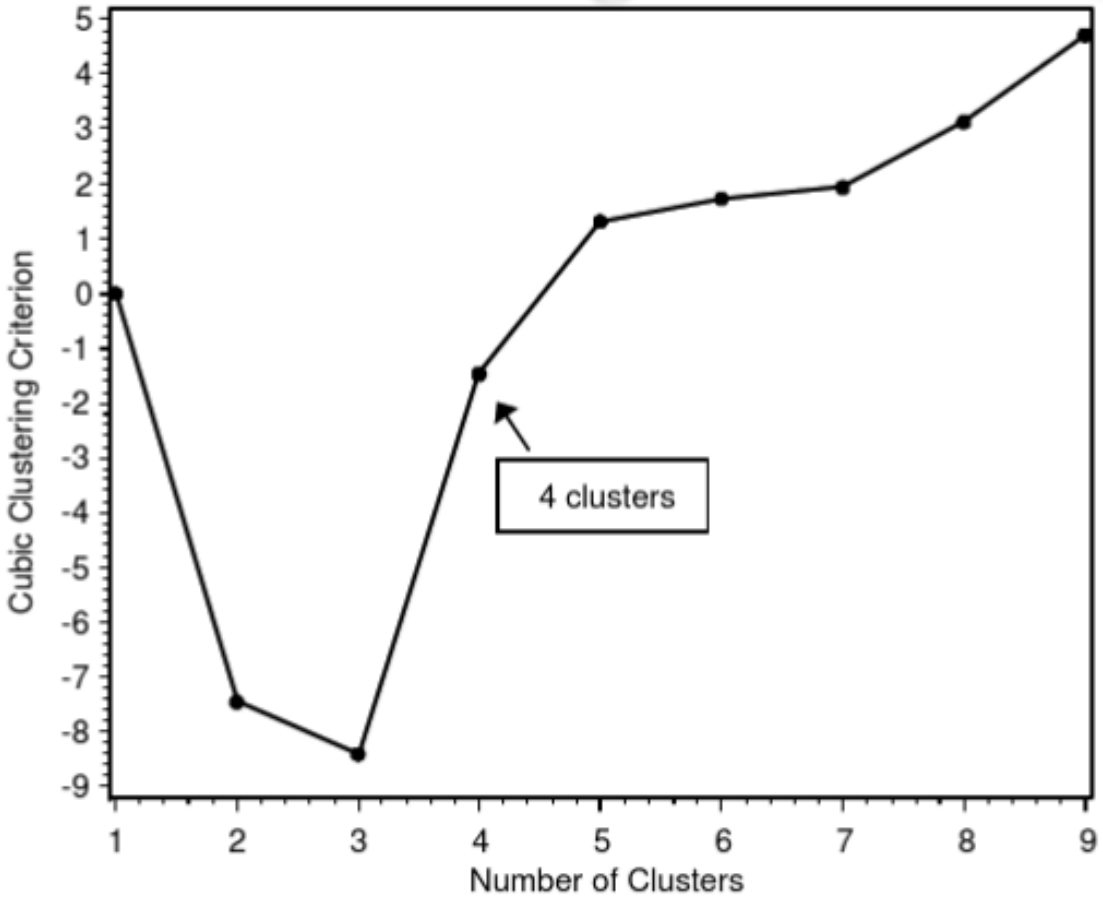


Figure 9.6 CCC as a function of the number of clusters.

- Image Source: Tufféry, Stéphane. *Data Mining and Statistics for Decision Making*. Wiley *2011*

Determining Number of Clusters with the Cubic Clustering Criterion (CCC)

WARNING: Do not expect the CCC to be common knowledge outside of the SAS domain.

[CITATION] SAS technical report A-108
WS **Sarle** - The Cubic Clustering Criterion. Cary, NC: SAS Institute, **1983**
Cited by 15 Related articles Cite Save

‘Overall R-Squared’ and ‘Pseudo-F’

These statistics draw connections between a final clustering and ANOVA.

- Total Sum of Squares (SST)
- Between Group Sum of Squares (SSB)
- Within Group Sum of Squares (SSW)
  - This is the k-means objective previously referred to as SSE.
  - Minimizing SSW => Maximizing SSB
- SST = SSB + SSW.
- ‘Overall R^2’ = SSB/SST
- $$pseudo\,F = \frac{R^2(k-1)}{(1-R^2)/(n-k)}$$

Example: PenDigit Data

- Goal: Automatic recognition of handwritten digits
- Digit database of 250 samples from 44 writers
- Subjects wrote digits in random order inside boxes of 500 by 500 tablet pixel resolution
- Spatial resampling to obtain a constant number of regularly spaced points on the trajectory
  - (x_1, x_2) give the first point coordinate
  - (x_3, x_4) give the second point coordinate
  - etc.

Example: PenDigit Data

```
proc fastclus   data=datasets.pendigittest
                maxclusters=10
                out = clus;
     var x1--x16;
run;
```

Example: PenDigit Data

Cluster Summary						
Cluster	Frequency	RMS Std Deviation	Maximum Distance from Seed to Observation	Radius Exceeded	Nearest Cluster	Distance Between Cluster Centroids
1	192	17.9838	131.7		7	108.9
2	880	20.9814	127.8		8	83.1740
3	115	16.3779	115.2		10	110.4
4	133	15.6579	118.4		5	101.5
5	575	19.9900	121.8		6	100.6
6	586	20.2352	129.9		5	100.6
7	217	15.5165	115.1		1	108.9
8	357	20.3639	124.4		2	83.1740
9	138	17.8981	119.2		2	100.3
10	305	15.4317	132.1		3	110.4

The first step to creating your own hierarchical dendrogram.

Example: PenDigit Data

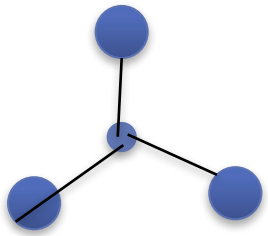
Statistics for Variables				
Variable	Total STD	Within STD	R-Square	RSQ/(1-RSQ)
X1	35.99529	23.35138	0.580229	1.382250
X2	14.73708	11.63537	0.378247	0.608354
X3	26.48636	19.87561	0.438336	0.780422
OVER-ALL	30.24017	19.27952	0.594581	1.466581

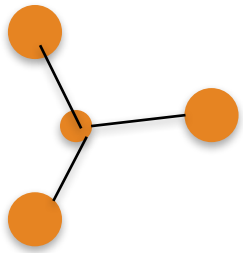
```
proc glm data= clus;
class cluster;
model x1 = cluster;
run; quit;
```

Example: PenDigit Data

Statistics for Variables				
Variable	Total STD	Within STD	R-Square	RSQ/(1-RSQ)
X1	35.99529	23.35133	0.580229	1.382250
X2	14.73708	11.63537	0.378247	0.608354
X3	26.48636	19.87561	0.438336	0.780422
OVER-ALL	30.24017	19.27952	0.594581	1.466581

$$\frac{\sum_{C_j} \sum_{x_i \in C_j} \|\mathbf{x}_i - \mathbf{c}_j\|^2}{\sum_{x_k} \|\mathbf{x}_k - \bar{\mathbf{x}}\|^2}$$

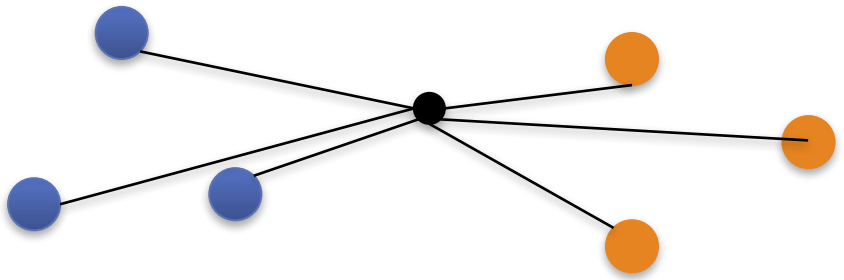




Example: PenDigit Data

Statistics for Variables				
Variable	Total STD	Within STD	R-Square	RSQ/(1-RSQ)
X1	35.99529	23.35133	0.580229	1.382250
X2	14.73708	11.63537	0.378247	0.608354
X3	26.48636	19.87561	0.438336	0.780422
OVER-ALL	30.24017	19.27952	0.594581	1.466581

$$\frac{\sum_{C_j}\sum_{x_i \in C_j} \|\mathbf{x}_i - \mathbf{c}_j\|^2}{\sum_{x_k} \|\mathbf{x}_k - \bar{\mathbf{x}}\|^2}$$



Example: PenDigit Data

Statistics for Variables				
Variable	Total STD	Within STD	R-Square	RSQ/(1-RSQ)
X1	35.99529	23.35133	0.580229	1.382250
X2	14.73708	11.63537	0.378247	0.608354
X3	26.48636	19.87561	0.438336	0.780422
OVER-ALL	30.24017	19.27952	0.594581	1.466581

Essentially using the centroids as predictions and then computing R-squared.