SI 618 – Week 11

Instructor: Ceren Budak

Some slides from: Yuhang Wang and Kevyn Collins-Thompson

A few point about the projects

- Must be in R.
- Focus on *exploratory* data analysis. Broader machine learning methods better suited for other classes.
  - But applying machine learning to perform classification and then doing exploratory data analysis to examine feature importance etc. would be a good match as well.
- Some proposals list a number of similar questions that can be classified under the same umbrella. This is great, but treat that as one type of analysis. (e.g. the relationship between different types of demographic characteristics and an outcome variable).
- We mentioned a few ideas you proposed as promising directions. That doesn't mean you have to take them on. It also doesnt mean the exact analysis you have there is the exact right approach. We wanted to give you a direction but always good to explore data in different ways.

Types of machine learning

- Unsupervised learning
  - Finding structure in unlabeled data
- Supervised learning
  - Making predictions based on labeled data
  - Predictions like regression or classification
- Reinforcement learning

Labeled vs. unlabeled data

Feature

Label

Observations

Color	Shape	Size (cm)	Group
Blue	Square	10	1
Red	Ellipse	2.4	1
Red	Ellipse	20.7	2

Sample from Murphy, Machine Learning: A Probabilistic Perspective

Unsupervised Methods

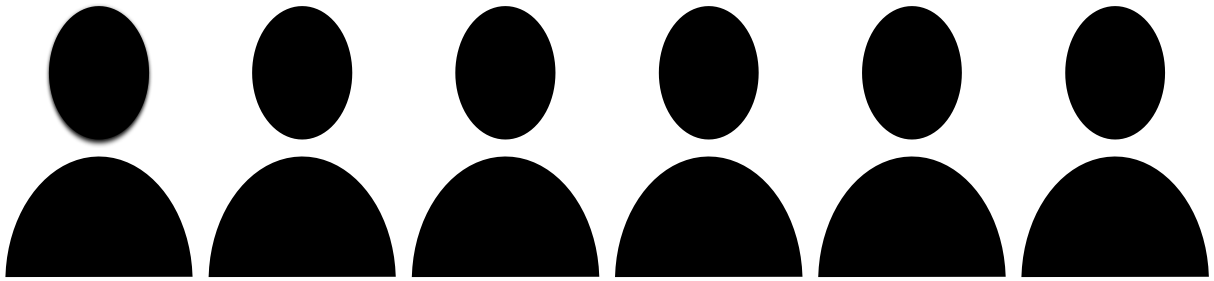
- Examples of tasks:
  - Finding homogenous subgroups within larger group
    - Clustering (this lecture)

  - Finding patterns in the features of the data
    - Dimensionality Reduction (next lecture)
  - …

- Challenges and Advantages:
  - No single goal of analysis
  - Requires more creativity
  - Much more unlabeled data available than cleanly labeled data

Unsupervised learning

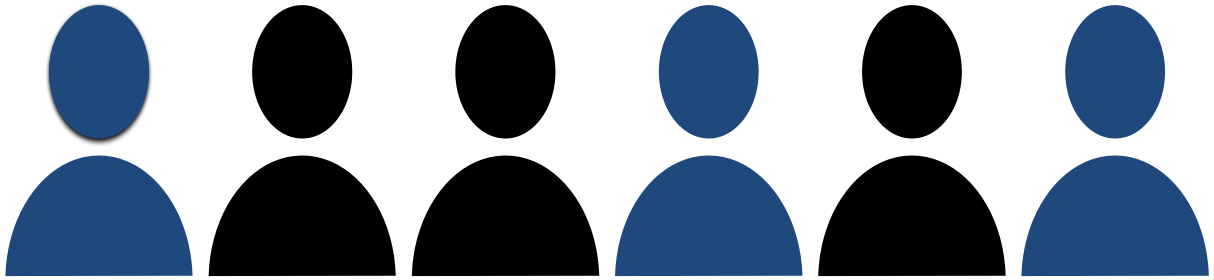
- Finding homogenous subgroups within larger group



People have features such as income, education attainment, and gender

Unsupervised learning

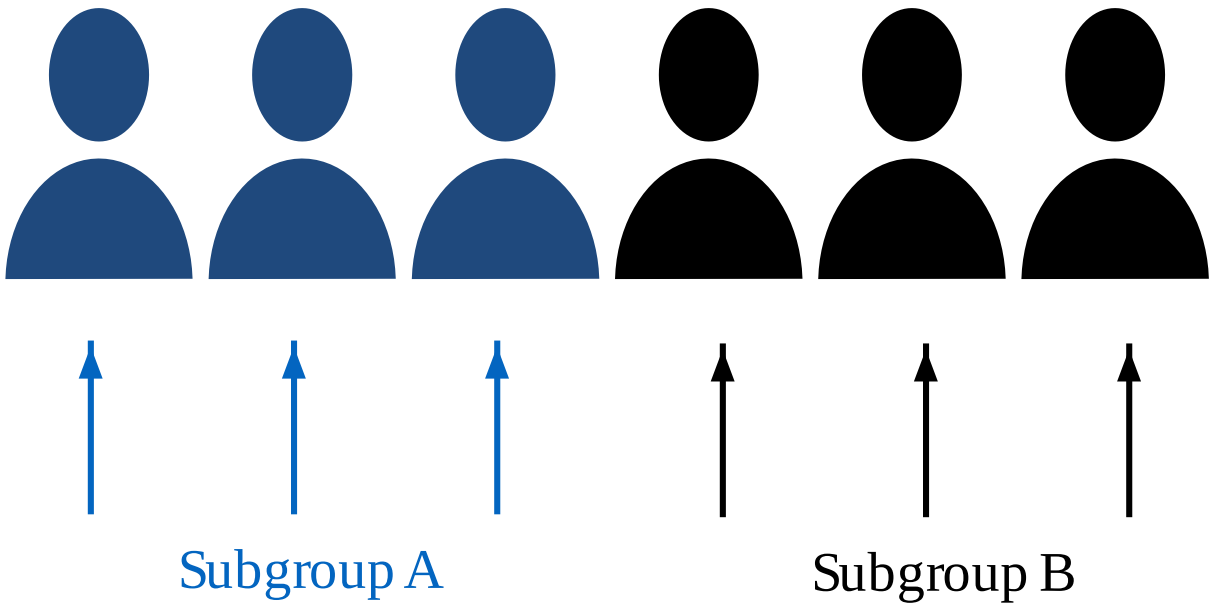
- Finding homogenous subgroups within larger group



People have features such as income, education attainment, and gender

Unsupervised learning

- Finding homogenous subgroups within larger group



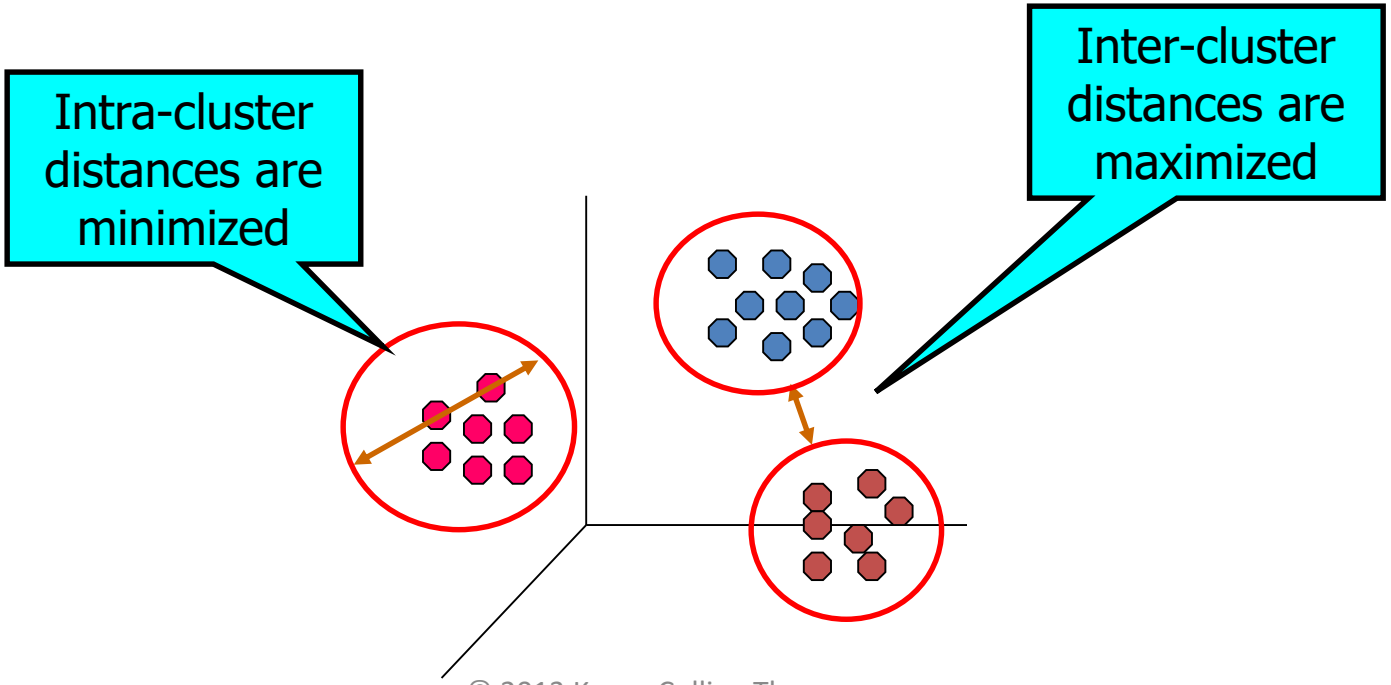
We can use clustering to tackle this problem

Clustering as an exploratory data analysis tool

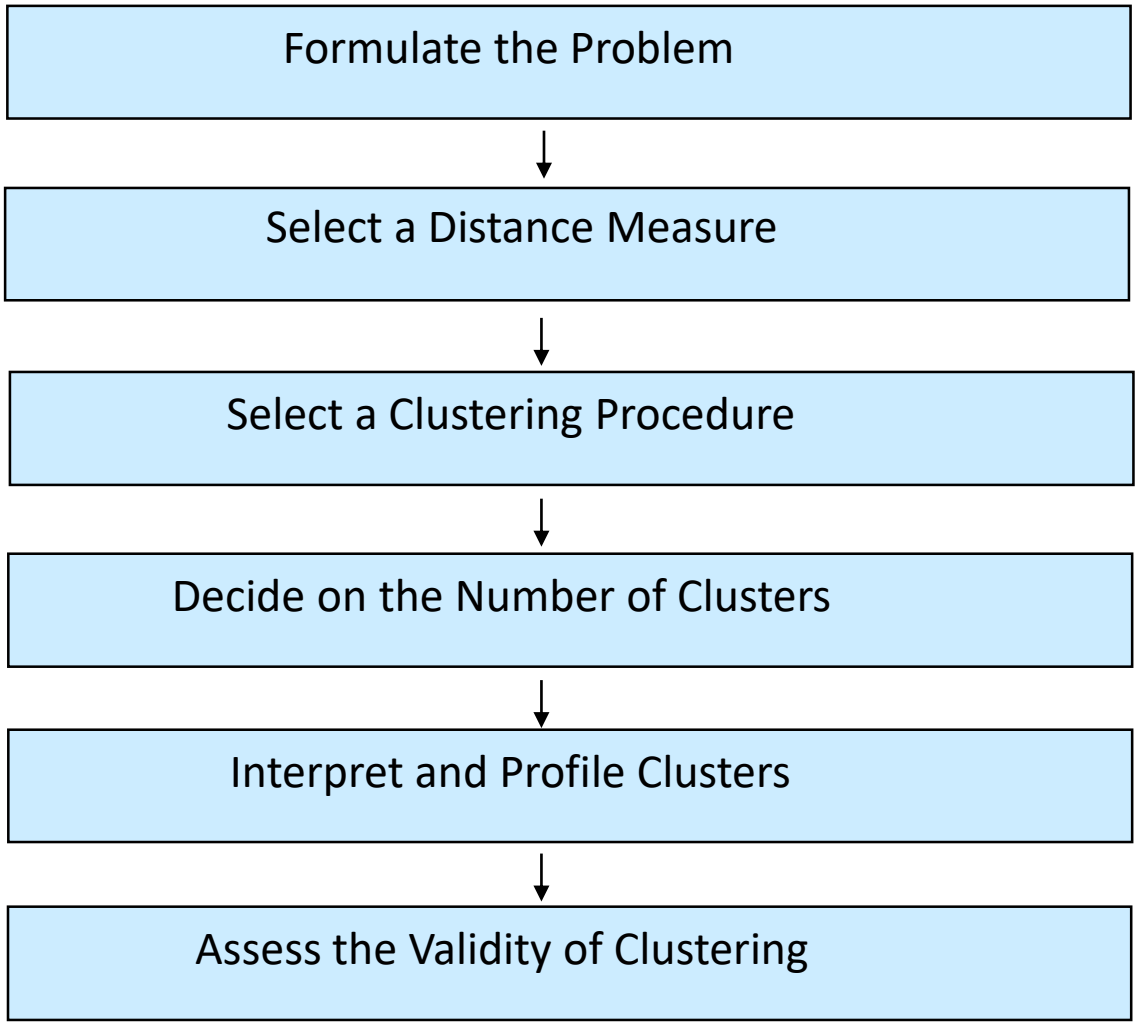
- Data understanding
  - Finding underlying factors, groups, structure
- Data navigation
  - Web search and browsing
- Data reduction
  - Clustering creates a new nominal level variable that can be used in any further analysis.
  - In one-dimension, a good way to quantize real-valued variables into k non-uniform buckets
- Data smoothing
  - Infer missing attributes from cluster neighbors

Cluster analysis finds 'interesting' groups of objects based on similarity

- What typically makes a 'good' clustering?
  - Members are highly similar to each other
    - Minimize within-cluster distances
  - Well-separated from other clusters
    - Maximize between-cluster distances



Summary: Conducting Cluster Analysis

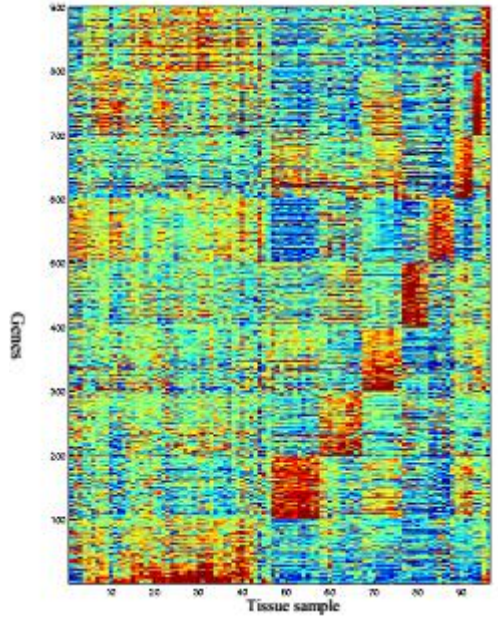


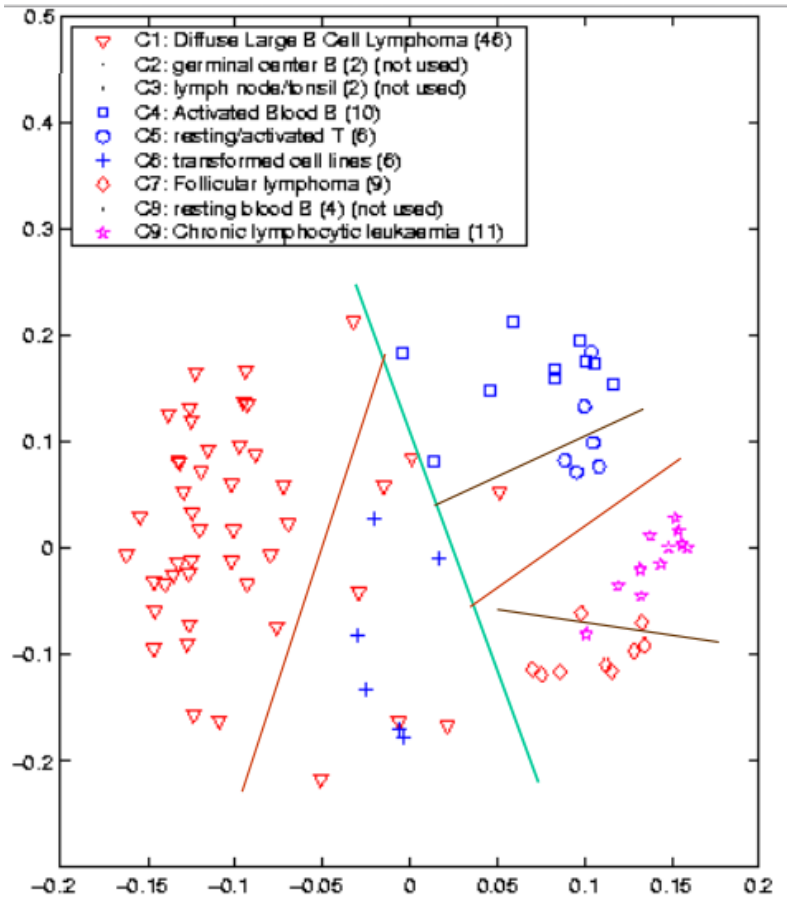
Clustering arises naturally in many fields

- Health
  - DNA gene expression
    - Cluster cancer variants into treatment groups, based on immunomarkers of cell samples
  - Medical imaging
    - Find likely tumors
- Business
  - Market segments
  - Web site visitors
- Social network analysis
  - Find communities
- Information Retrieval:
  - Search results clustered by similarity, event or topic
  - Personalization for groups of similar users
- Speech understanding
  - Convert waveforms into one of k categories (known as Vector Quantization)

Example: Clustering lymphoma cancer tissue samples

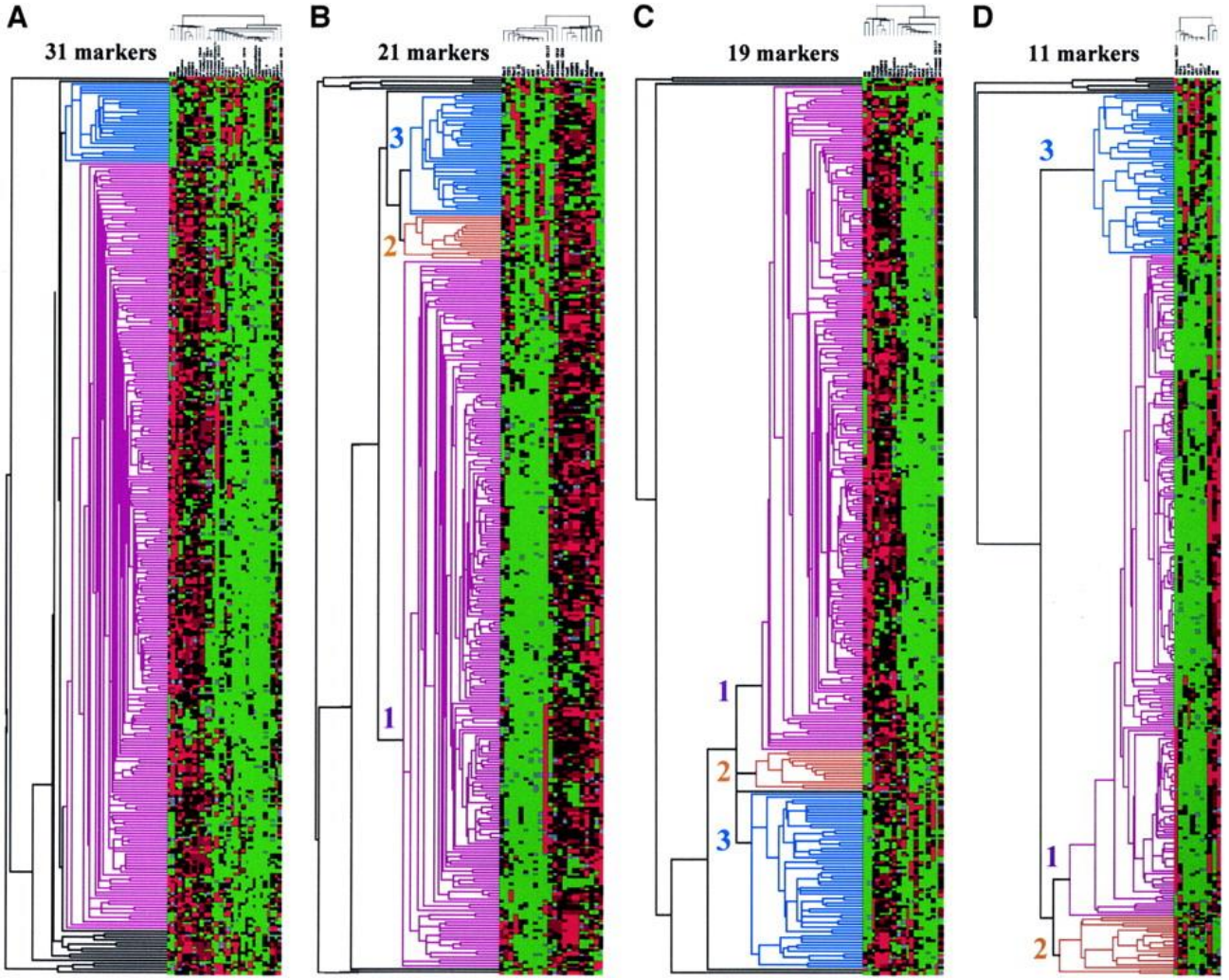
- B-cell lymphoma go through different stages
- Can we detect which stage automatically?





[Source: Chris Ding, ICML 2004 Tutorial on Spectral Clustering]

Hierarchical clustering analysis with 31(A), 21(B), 19(C), and 11(D) immunomarkers. Groups breast cancer cases (rows) into clinically relevant classes with similar immunomarkers (columns)



Makretsov N A et al. Clin Cancer Res 2004;10:6143-6151

Clustering can be ambiguous: What is the 'best' clustering here?

How many clusters?

Six Clusters

Two Clusters

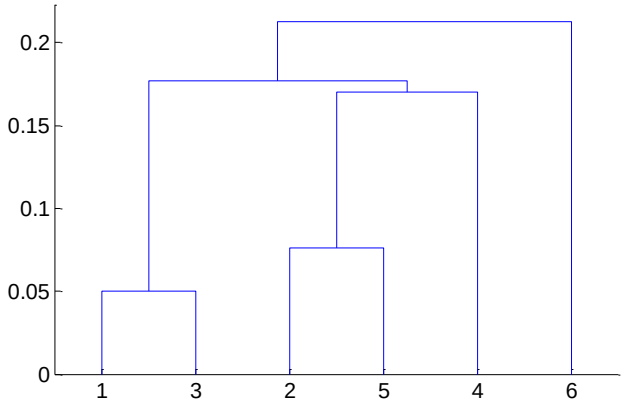
Four Clusters

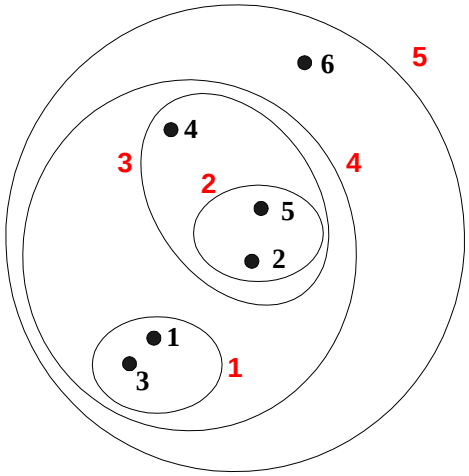
What algorithms are used to find clusters? Answer: huge range of approaches

- Assigning objects to clusters
  - 'Hard' (partitional) each object belongs to exactly 1 cluster
  - 'Soft' : each object can belong to multiple clusters
- Hierarchical vs non-hierarchical
  - A set of nested clusters organized as a tree
- By far most widely-used fall into two types:
  - **Hierarchical**: agglomerative, top-down, etc.
  - **Partitional**: k-means, etc.

Hierarchical Clustering

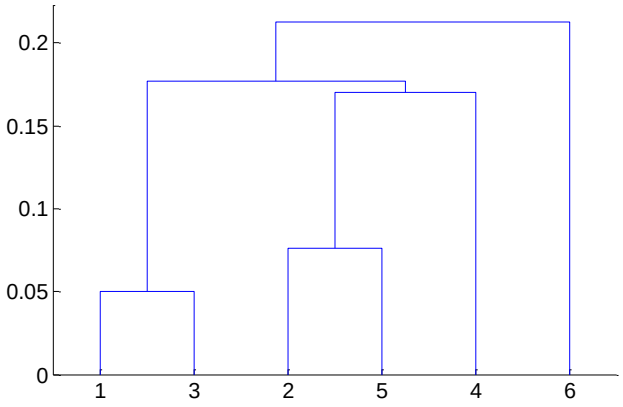
- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)



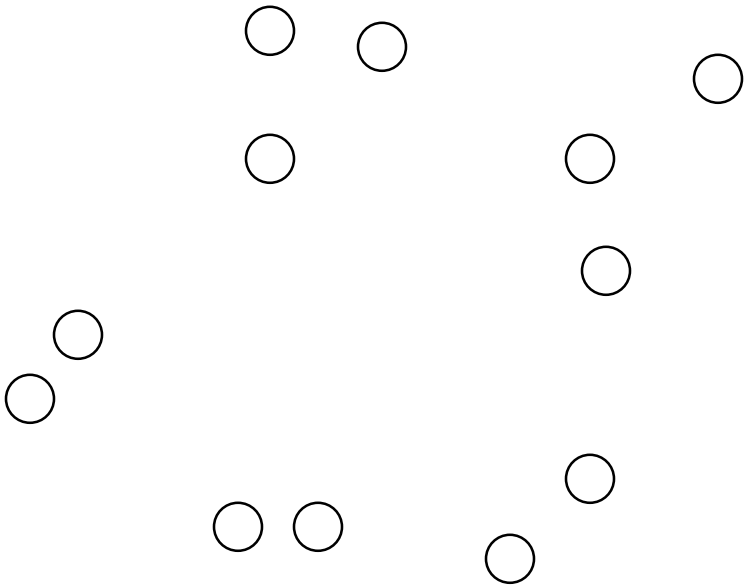
Hierarchical clustering

- Bottom-up ('Agglomerative')
  - Start with each point being in its own cluster
  - At each step
    - Merge the most similar pair of clusters based on a cost function
    - Continue until you have k clusters, or everything is in one big cluster
- Top-down ('Divisive')
  - Start with all points in a single big cluster
  - At each step:
    - Split the cluster into two smaller clusters based on a cost function
    - Continue until you have k clusters, or each point is in its own cluster

http://wiki.stat.ucla.edu/socr/index.php/SOCR_EduMaterials_AnalysisActivities_HierarchicalClustering

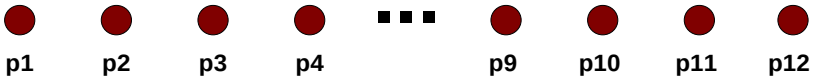
Agglomerative (bottom-up) Clustering: Starting Situation

- Start with clusters of individual points and a proximity matrix of object-to-object distances



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

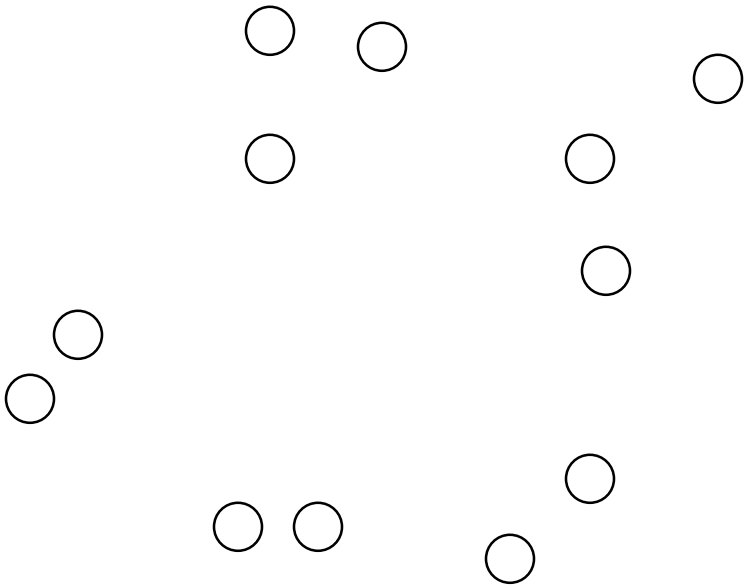


Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity matrix
  6. **Until** only a single cluster remains
- Key operation: computation of the proximity of two clusters. The cost function.
  - Different approaches to defining the distance between clusters distinguish the different algorithms

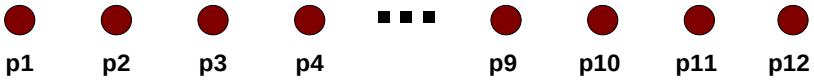
Agglomerative (bottom-up) Clustering: Starting Situation

- Start with clusters of individual points and a proximity matrix of object-to-object distances



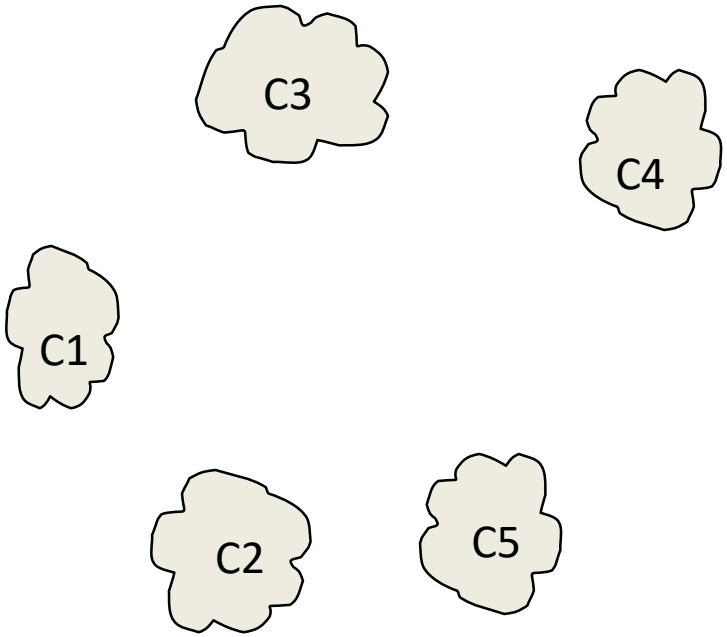
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix



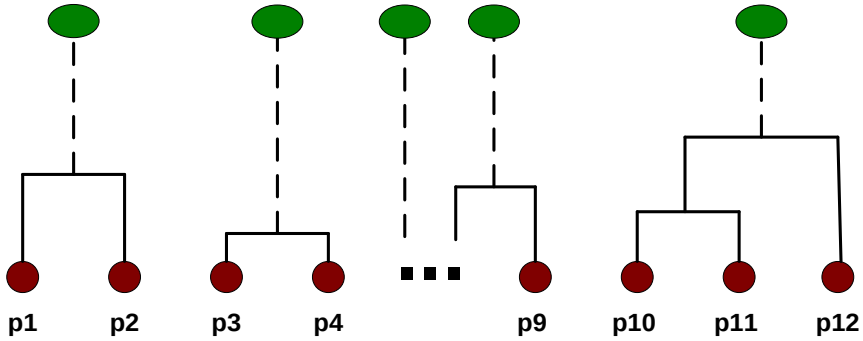
Intermediate Situation

- After some merging steps, we have some clusters



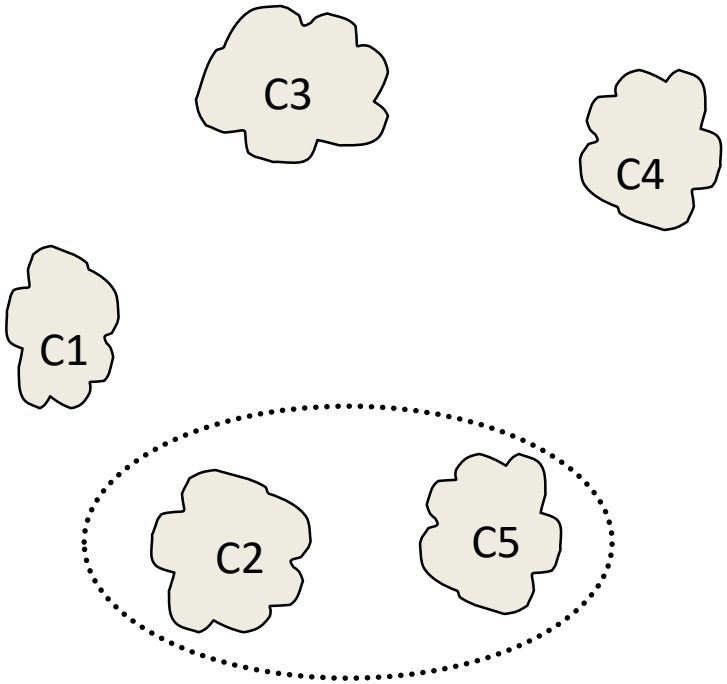
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



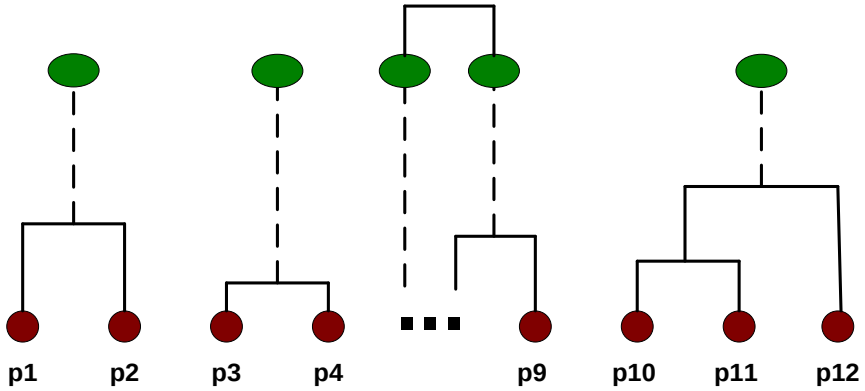
Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



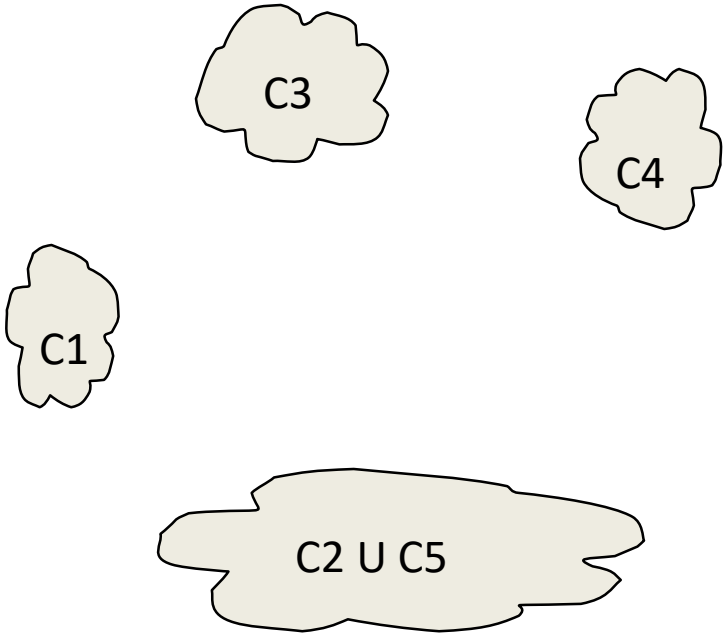
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



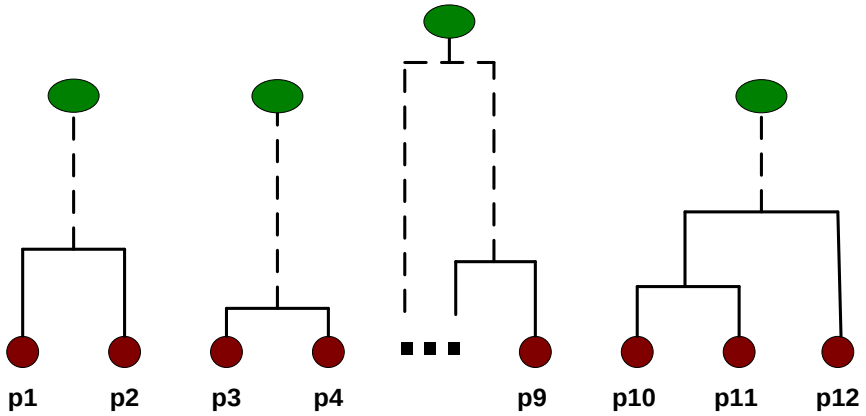
After Merging

- The question is "How do we update the proximity matrix?"

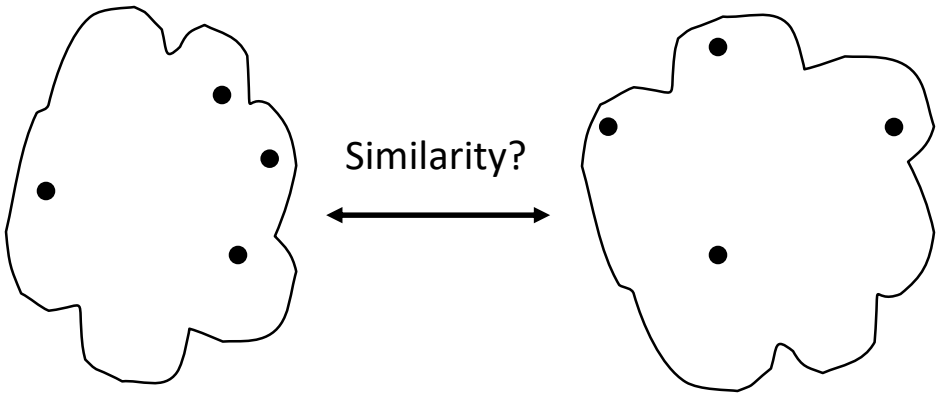


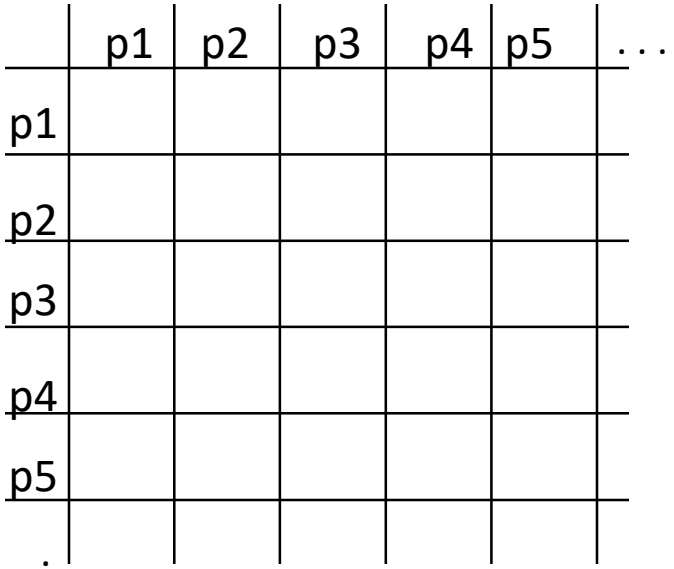
	C1	C2 + C5	C3	C4
C1		?		
C2 + C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

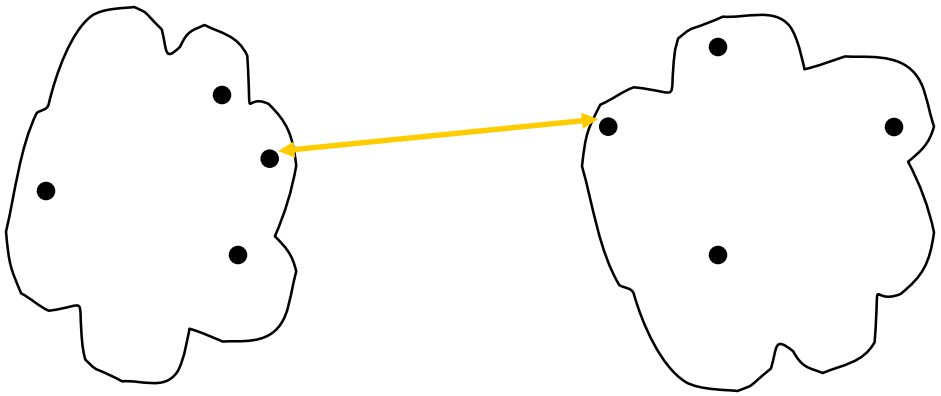




Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

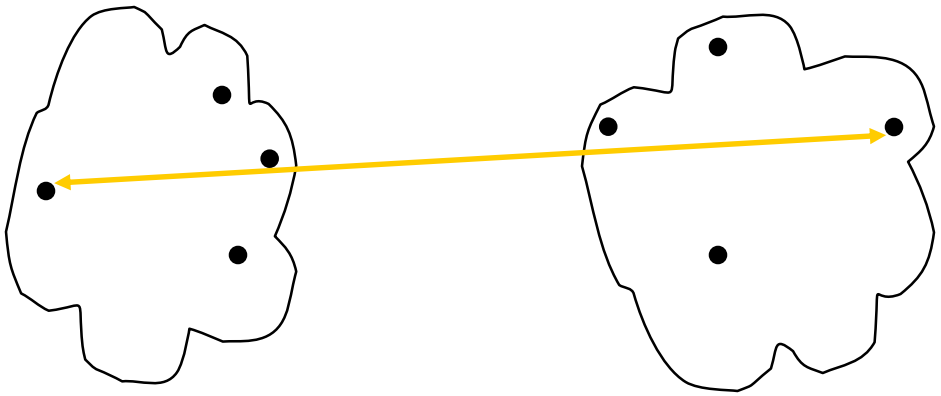


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

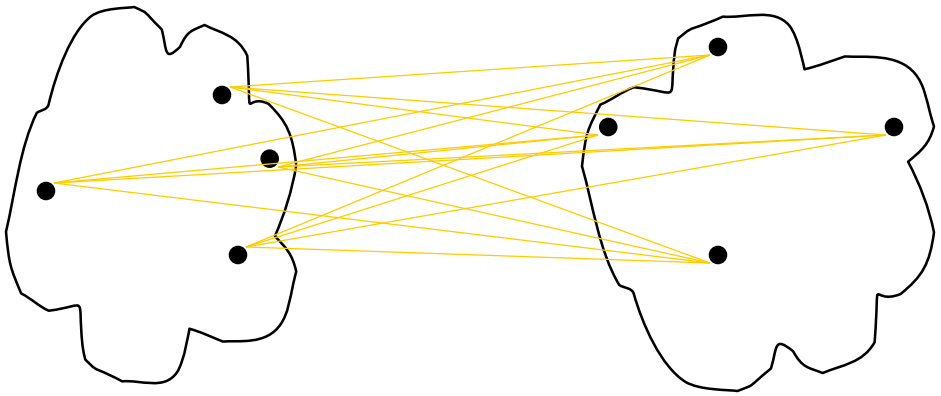


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

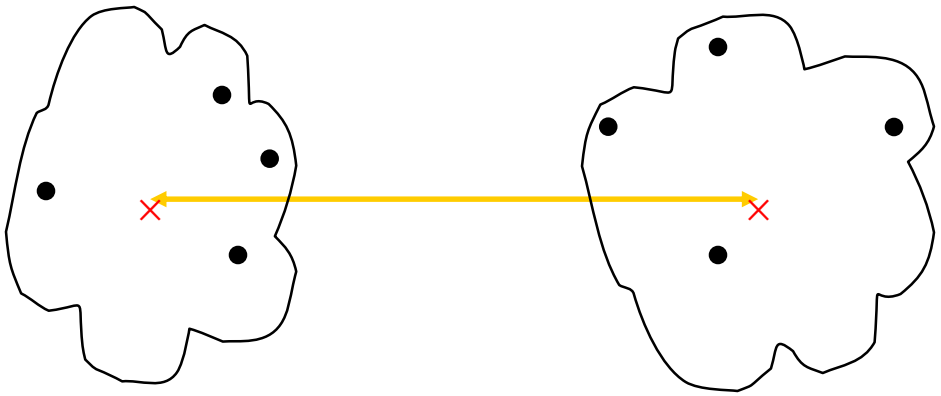


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity



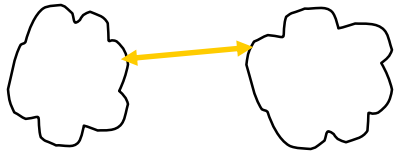
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

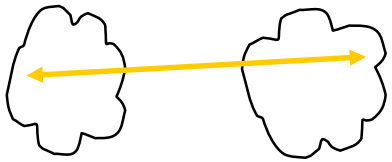
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

Cost functions for bottom-up (agglomerative) clustering

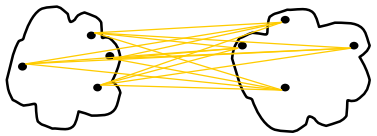
- Single linkage
  - Minimum distance between clusters



- Complete linkage
  - Max distance between clusters

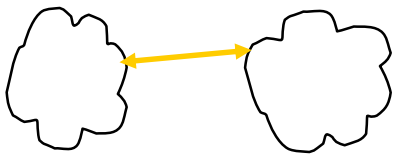


- Average linkage
  - Average distance between clusters

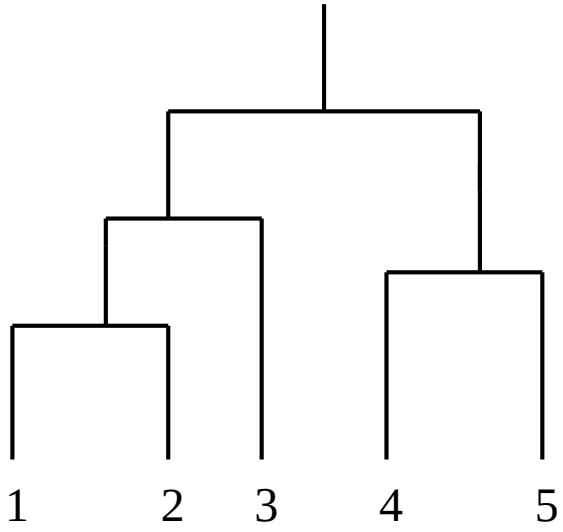


Cluster Similarity: MIN or Single Linkage

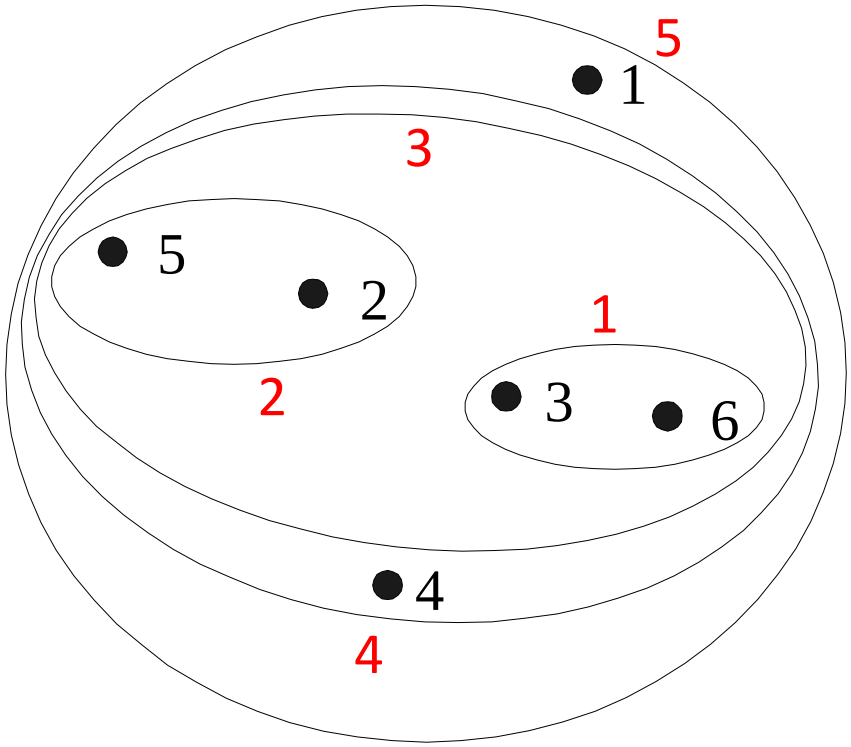
- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.



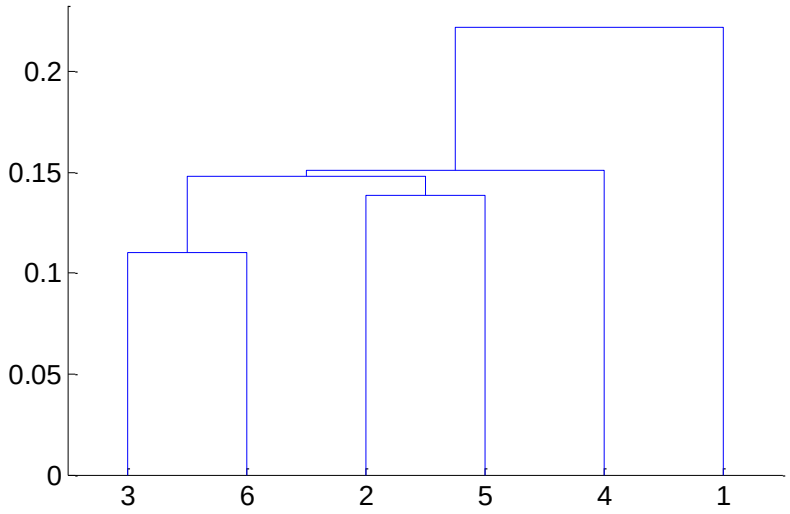
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MIN

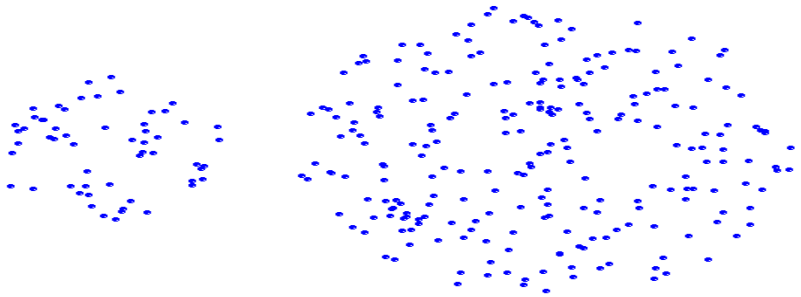


Nested Clusters

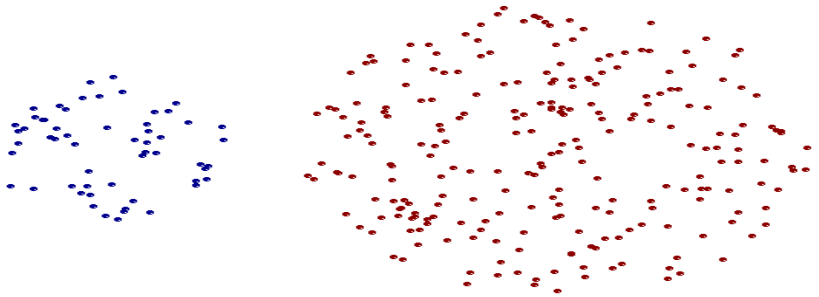


Dendrogram

Strength of MIN



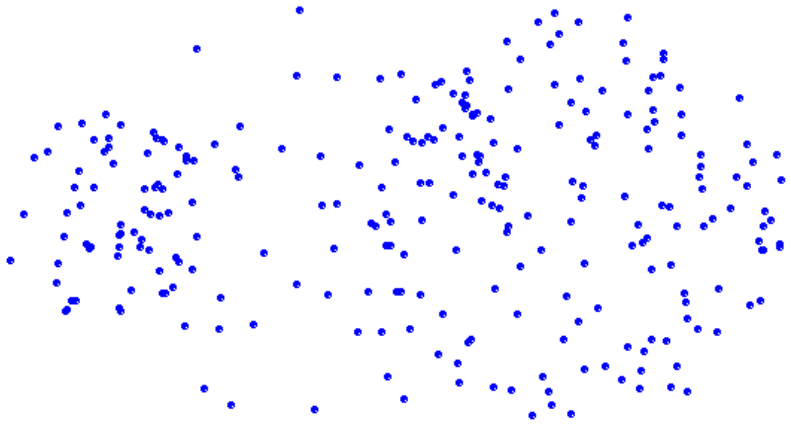
Original Points



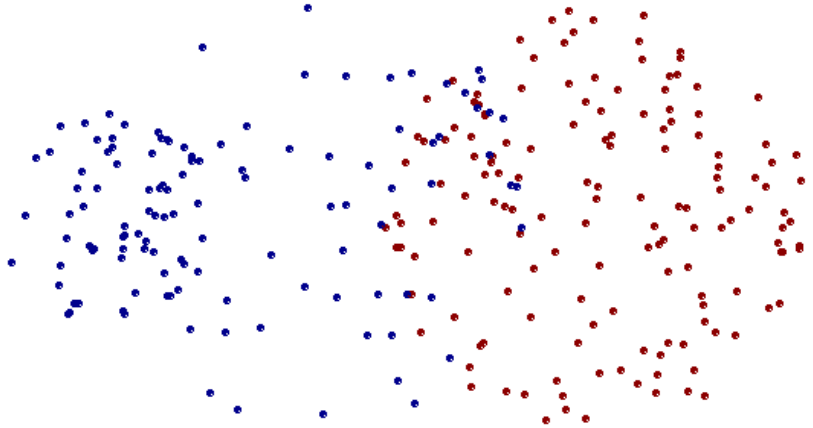
Two Clusters

- **Can handle non-elliptical shapes**

Limitations of MIN



Original Points

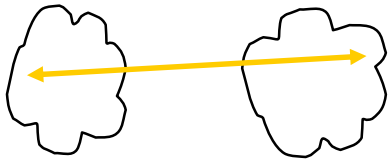


Two Clusters

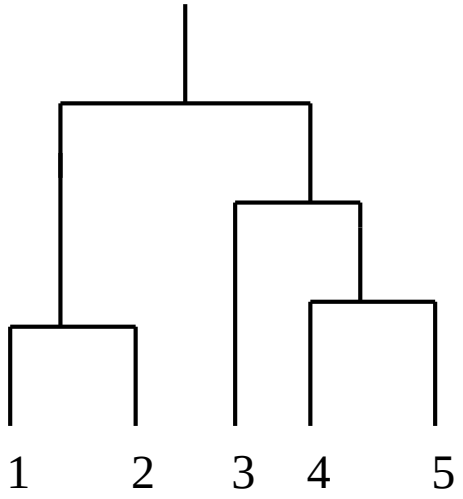
- **Sensitive to noise and outliers**
- **It produces long, elongated clusters**

Cluster Similarity: MAX or Complete Linkage

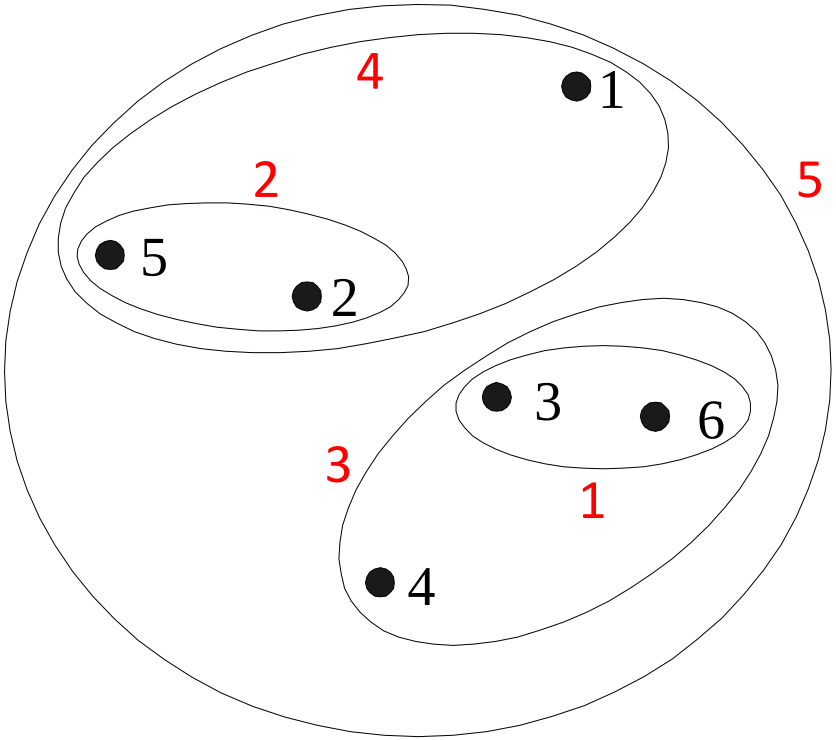
- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters



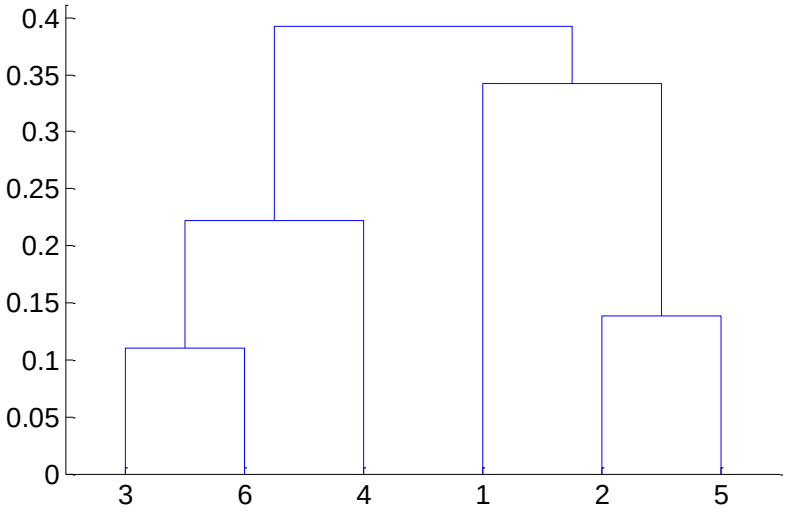
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MAX

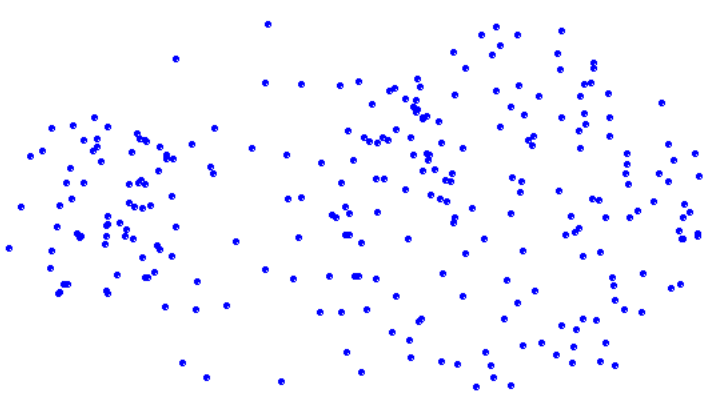


Nested Clusters

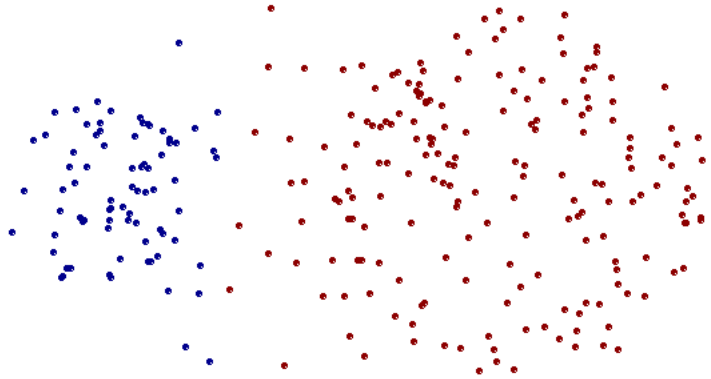


Dendrogram

Strength of MAX



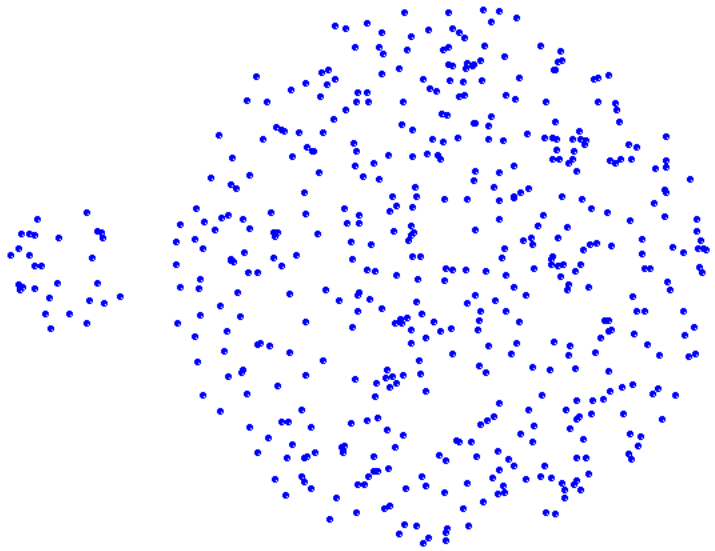
Original Points



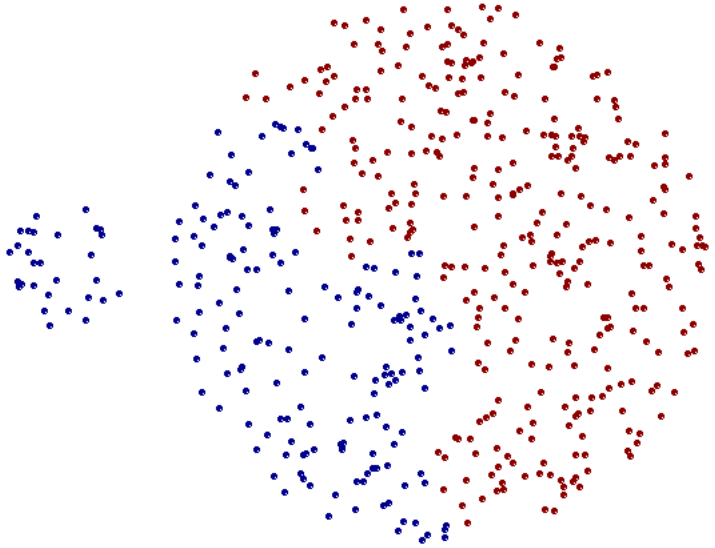
Two Clusters

- **More balanced clusters (with equal diameter)**
- **Less susceptible to noise**

Limitations of MAX



Original Points



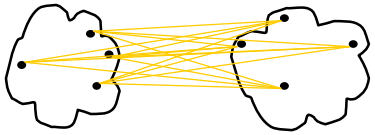
Two Clusters

- **Tends to break large clusters**
- **All clusters tend to have the same diameter – small clusters are merged with larger ones**

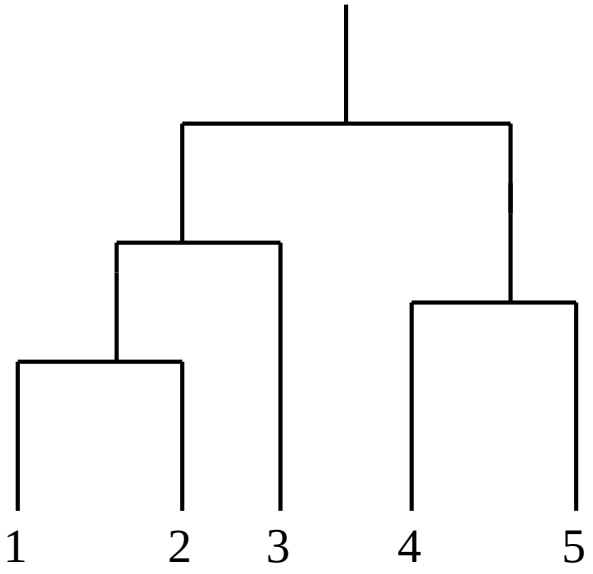
Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

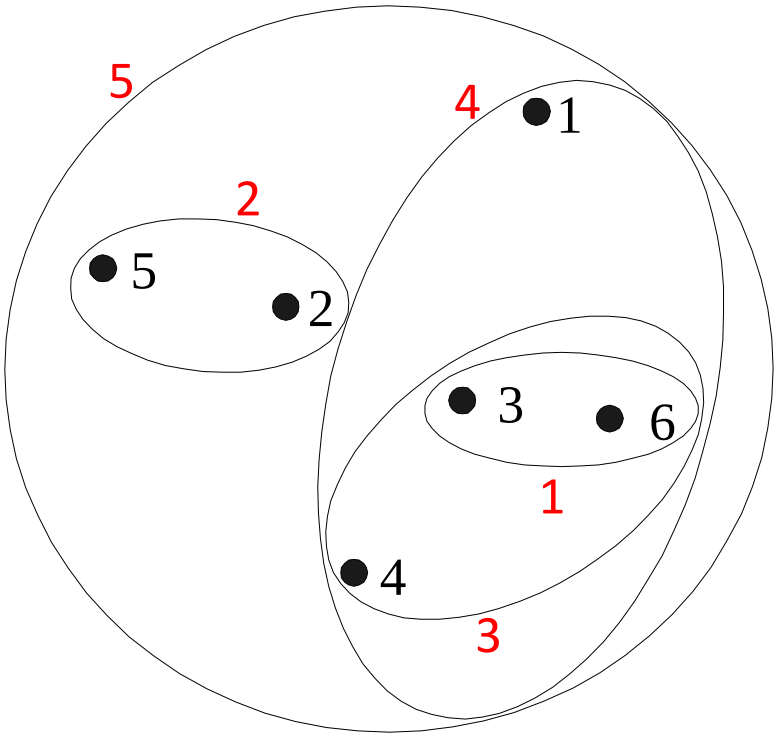
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$



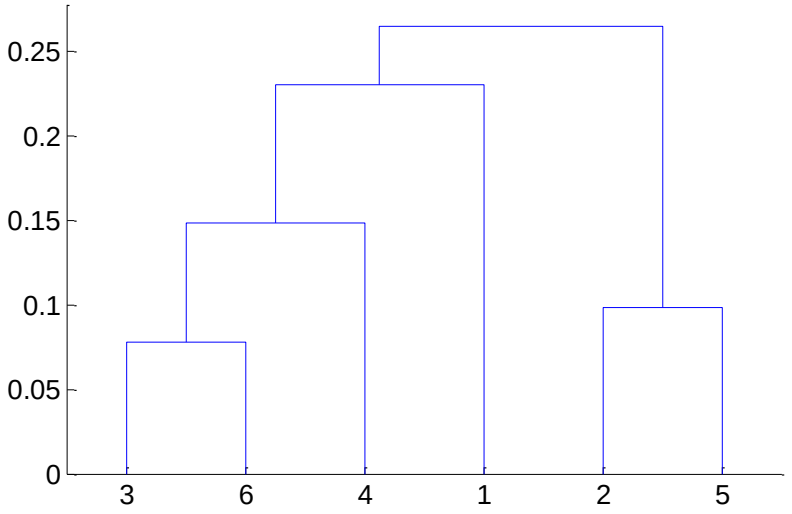
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: Group Average



Nested Clusters



Dendrogram

Hierarchical Clustering: Group Average

- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

Distance between two clusters

- **Centroid distance** between clusters C_i and C_j is the distance between the centroid r_i of C_i and the centroid r_j of C_j

$$D_{centroids}(C_i, C_j) = d(r_i, r_j)$$

Ward's method (1963)

- **Ward's distance** between clusters C_i and C_j is the ***difference*** between the ***total within cluster sum of squares for the two clusters separately***, and the ***within cluster sum of squares resulting from merging the two clusters*** in cluster C_{ij}

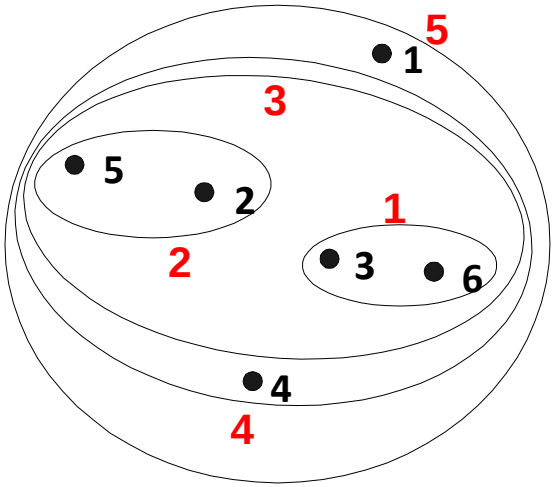
$$D_w(C_i, C_j) = \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 - \sum_{x \in C_{ij}} (x - r_{ij})^2$$

- r_i: centroid of C_i
- r_j: centroid of C_j
- r_{ij}: centroid of C_{ij}

Ward's distance for clusters

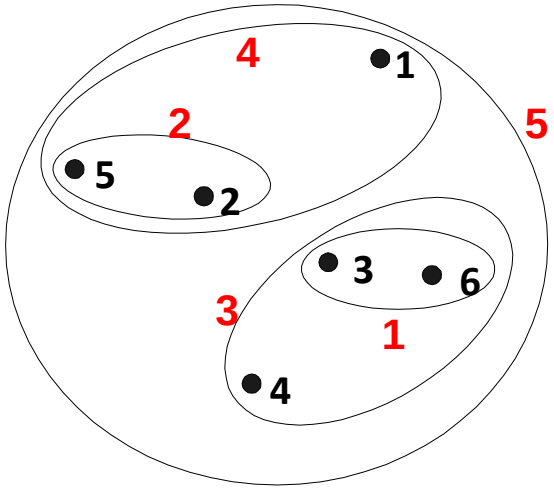
- Similar to group average and centroid distance
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of k-means

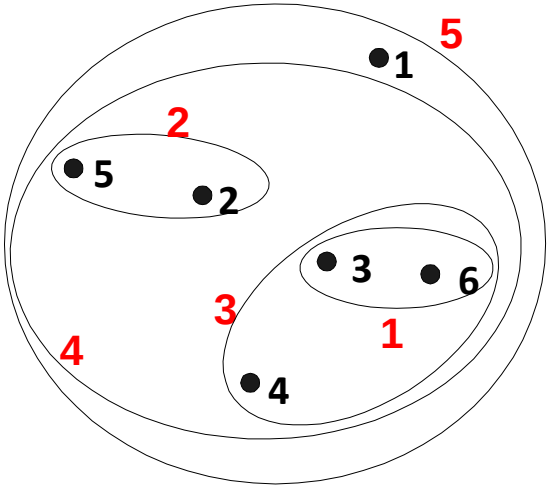
Hierarchical Clustering: Comparison



MIN

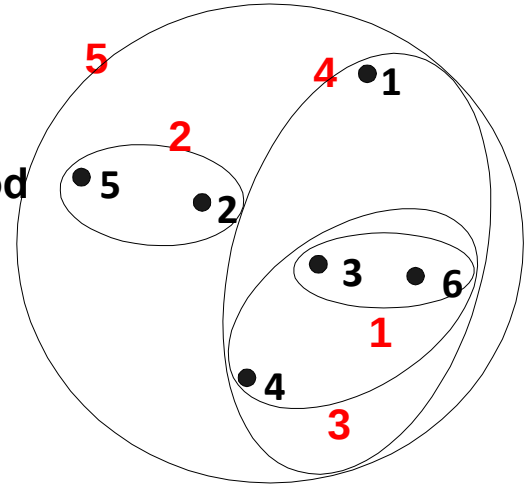
MAX





Group Average

Ward's Method



Which type of hierarchical clustering to use?

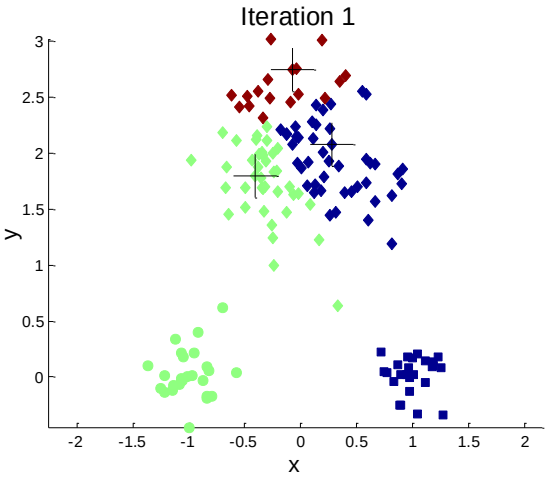
- Different methods have different strengths and weaknesses.
  - Ward's method tends to give equal sized clusters
  - Single linkage (nearest neighbor) tends to make long strings into a cluster.
  - Top-down is sensitive to early errors: bad first choice can wreck the entire process
  - Bottom-up can’t see the whole dataset

- Two major clustering algorithms
  - Hierarchical
  - K-means
- General questions:
  - How many clusters is best?
  - How can we assess and visualize cluster quality?
  - How can we visualize clusters?

K-means: the other massively popular clustering method.. and very different in nature

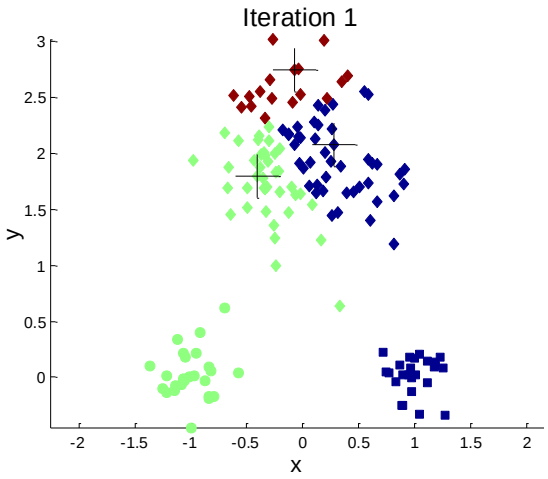
- Partitional clustering approach
- Each cluster associated with a centroid (centerpoint)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified in advance
- The basic algorithm is very simple

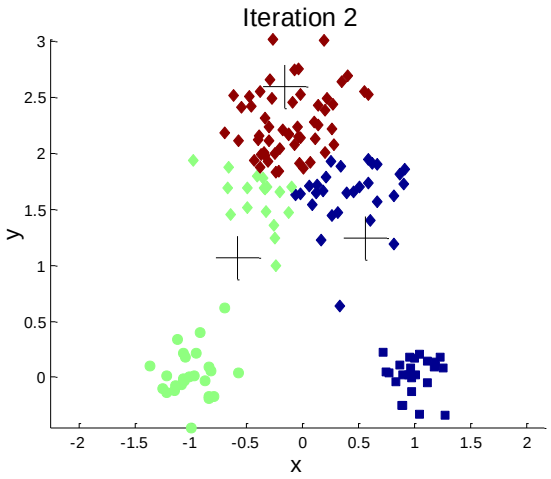
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

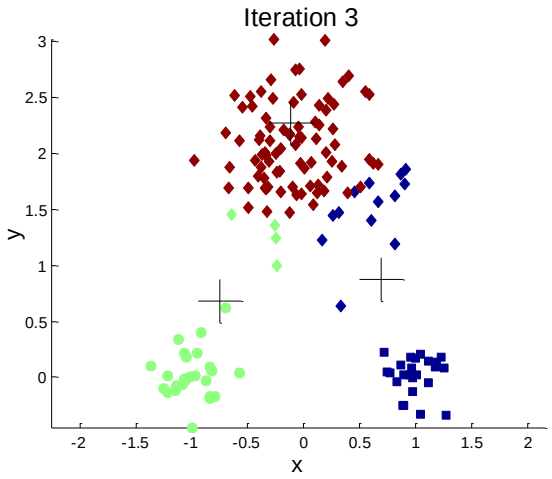


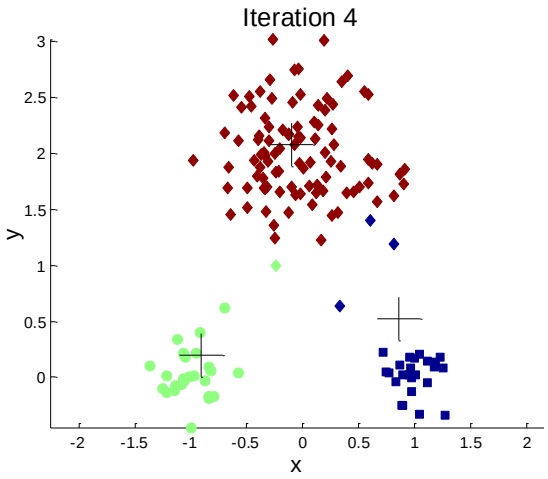
http://stat.ethz.ch/R-manual/R-devel/library/stats/html/kmeans.html

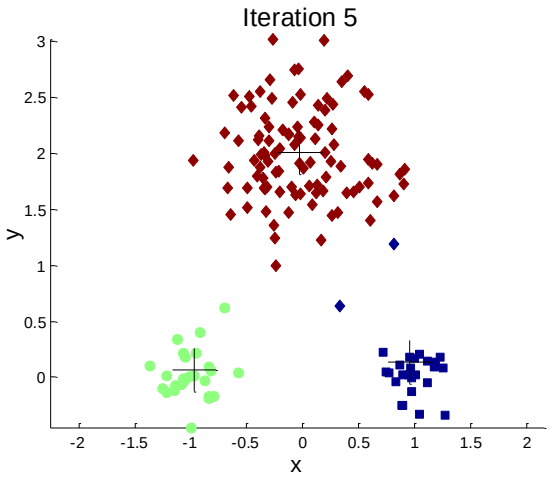
The k-means algorithm (k = 3)

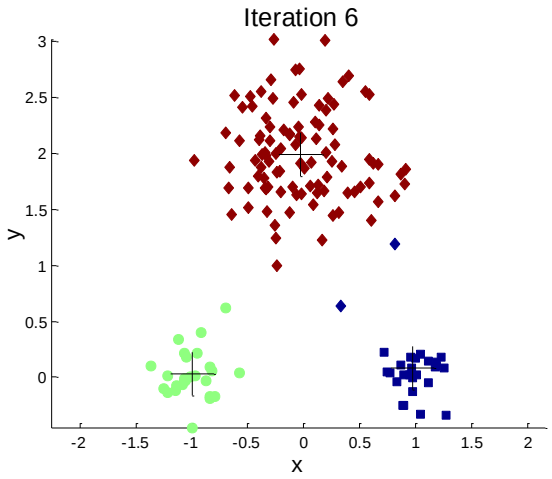












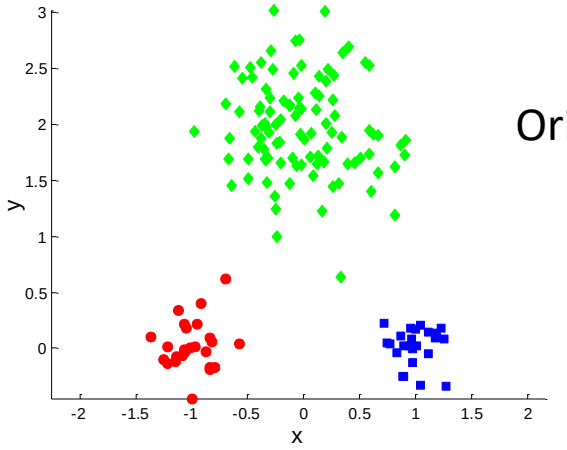
K-means is a special case of model-based clustering

- Assume data generated from **k** probability distributions
- ***Goal:*** find the distribution parameters
- ***Algorithm:*** Expectation Maximization (EM)
- ***Output:*** Distribution parameters and a **soft** assignment of points to clusters

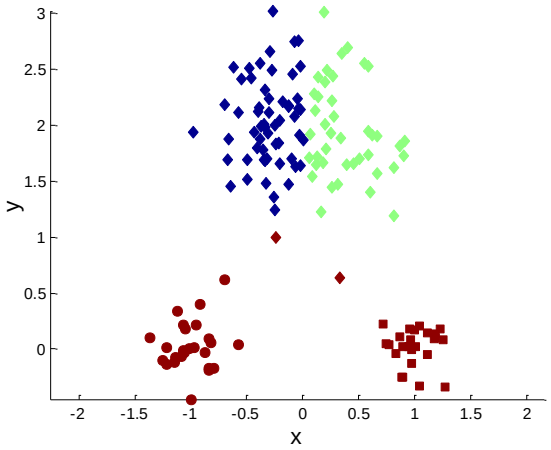
K-means Clustering – Details

- Different initializations can result in different solutions
  - Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
  - So multiple runs are sometimes done
- Centroid is typically the mean of the points in the cluster.
  - K-medoid: center must be an actual datapoint. Useful when mean of a feature is not defined or available
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most convergence happens in the first few iterations.
  - Often the stopping condition is changed to ‘Until relatively few points change clusters’
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

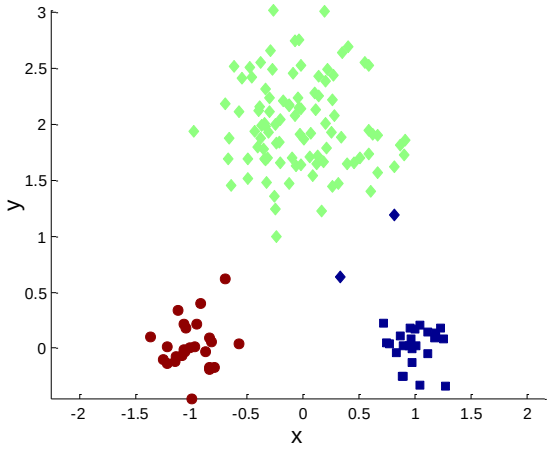
Two different K-means Clusterings



Original Points

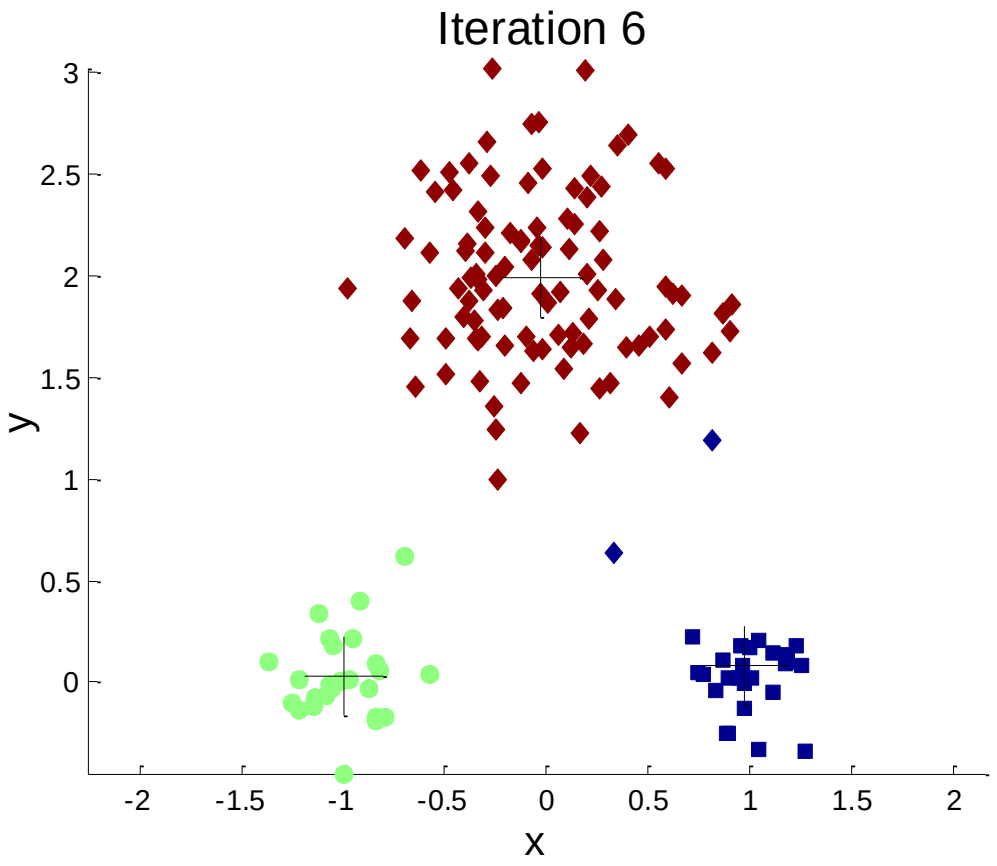


Optimal Clustering? No.

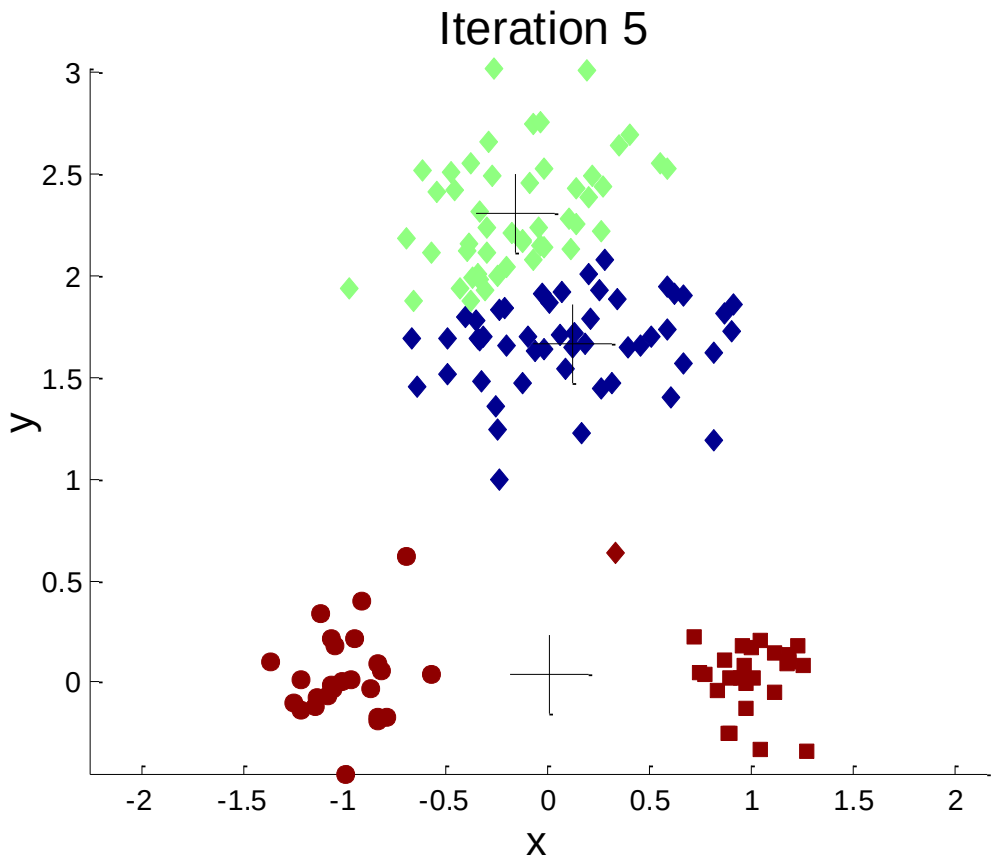


Optimal Clustering

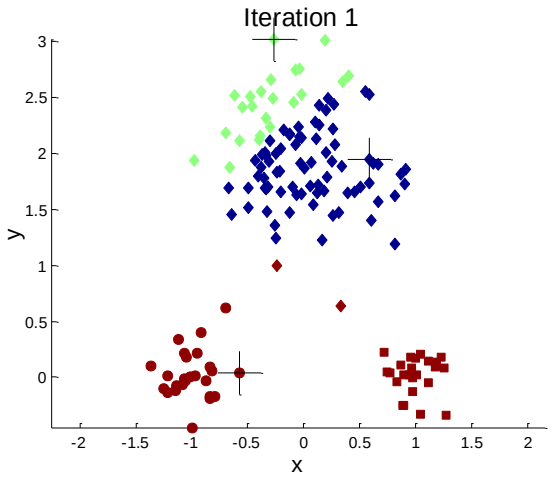
Importance of Choosing Initial Centroids

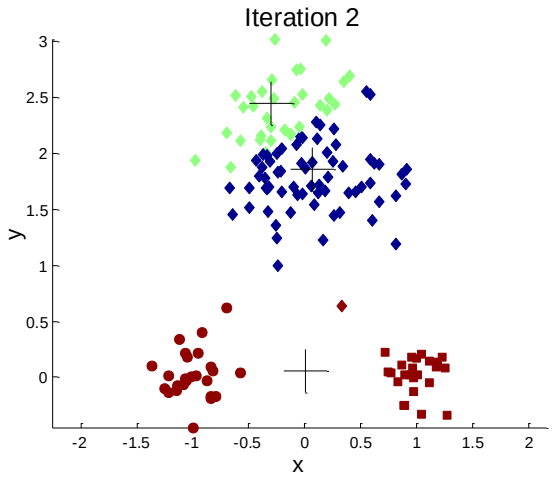


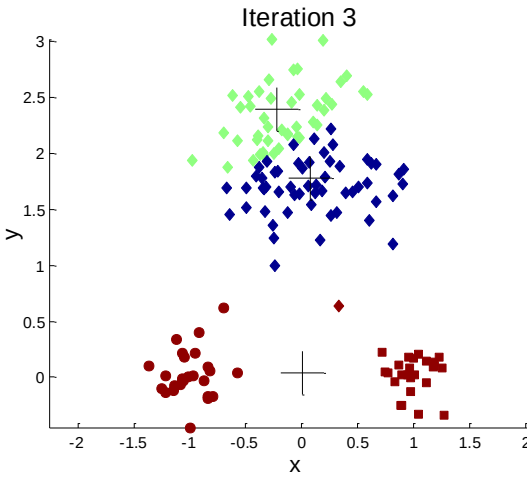
Importance of Choosing Initial Centroids …

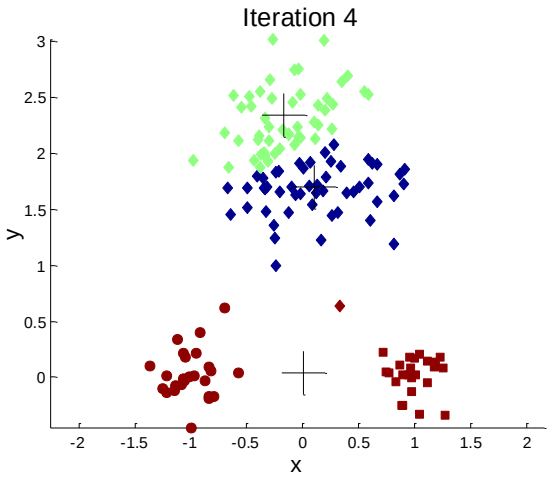


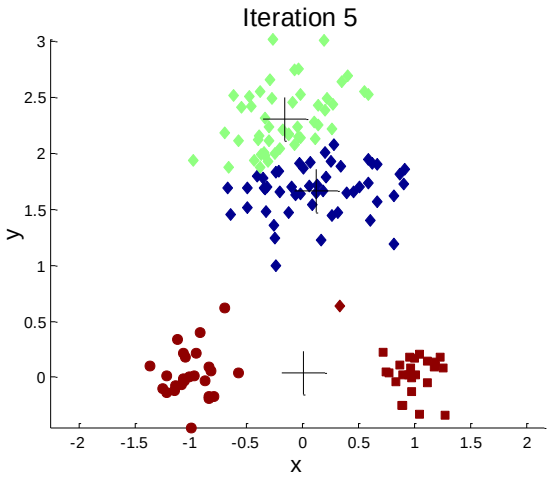
Importance of Choosing Initial Centroids …











Trying to find good optimal k-means clustering

- Idea 1: Be careful about where you start
  - Place first center on randomly chosen datapoint
  - Place second centroid on datapoint as far as possible from first center (or soft probabilistic version thereof)
  - Place *j*-th center on datapoint that's as far as possible from centers 1 thru *j* - 1
- Idea 2: Do many runs of k-means
  - Each from a different random start configuration
- Many heuristics around

Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers.

When should you use k-means vs hierarchical approach?

- Do you need to easily interpret the clusters?
- Do you know the right *k?*
- Hierarchal clustering is the sort that you might apply when there is a "tree" structure to the data (e.g. living things).
- K-means clustering does not assume a tree structure.
- If you have only two or three dimensions (or can sensibly reduce your data by factor analysis) you can plot it and see what sort of relationships you have. Are you looking for nice spherical clusters, or are long chains more suitable?
- k-means prefers solutions where clusters are of similar size
  - very different cluster sizes, shapes, densities can confuse it
  - complex cluster geometry, or outliers
  - need to specify and test for good *k* choice
- Can combine the two approaches, e.g.
  1. Try several hierarchical methods and see which gives the most interpretable clusters.
  2. Use k-means (with the hierarchical cluster centroids as starting points) to clean up the hierarchical cluster.

Clustering in R
Step 1: Data preparation

```
# Prepare Data
mydata <- na.omit(mydata) # listwise deletion of missing
mydata <- scale(mydata) # standardize variable scales
```

Note: Scaling is important. Think about what happens if points are clustered on one variable from 0-100 and another on 0.0-1.0

Reference: http://www.statmethods.net/advstats/cluster.html

Step 2: Clustering (if Hierarchical)

```
> head(cars.data)
                          MPG Weight Drive_Ratio Horsepower Displacement Cylinders
Buick Estate Wagon       16.9  4.360        2.73        155          350         8
Ford Country Squire Wagon 15.5  4.054        2.26        142          351         8
Chevy Malibu Wagon       19.2  3.605        2.56        125          267         8
Chrysler LeBaron Wagon   18.5  3.940        2.45        150          360         8
Chevette                 30.0  2.155        3.70         68           98         4
Toyota Corona            27.5  2.560        3.05         95          134         4

# Heirarchical clustering: compute distance matrix
cars.dist = dist(cars.data)

> as.matrix(cars.dist)
                     Buick Estate Wagon Ford Country Squire Wagon Chevy Malibu Wagon Chrysler LeBaron Wagon    Chevette Toyota Corona
Buick Estate Wagon                0.00000                13.125339          88.28867              11.30552 266.9576988    224.472053
Ford Country Squire Wagon        13.12534                 0.000000          85.78451              12.41165 264.0396368    222.397968
Chevy Malibu Wagon               88.28867                85.784507           0.00000              96.30480 178.7345577    136.657316
Chrysler LeBaron Wagon           11.30552                12.411652          96.30480               0.00000 274.8108417    232.809502
Chevette                        266.95770               264.039637         178.73456             274.81084   0.0000000     45.075897
Toyota Corona                   224.47205               222.397968         136.65732             232.80950  45.0758974      0.000000

cars.hclust <- hclust(cars.dist, method = "average")
```

Reference: http://www.statmethods.net/advstats/cluster.html

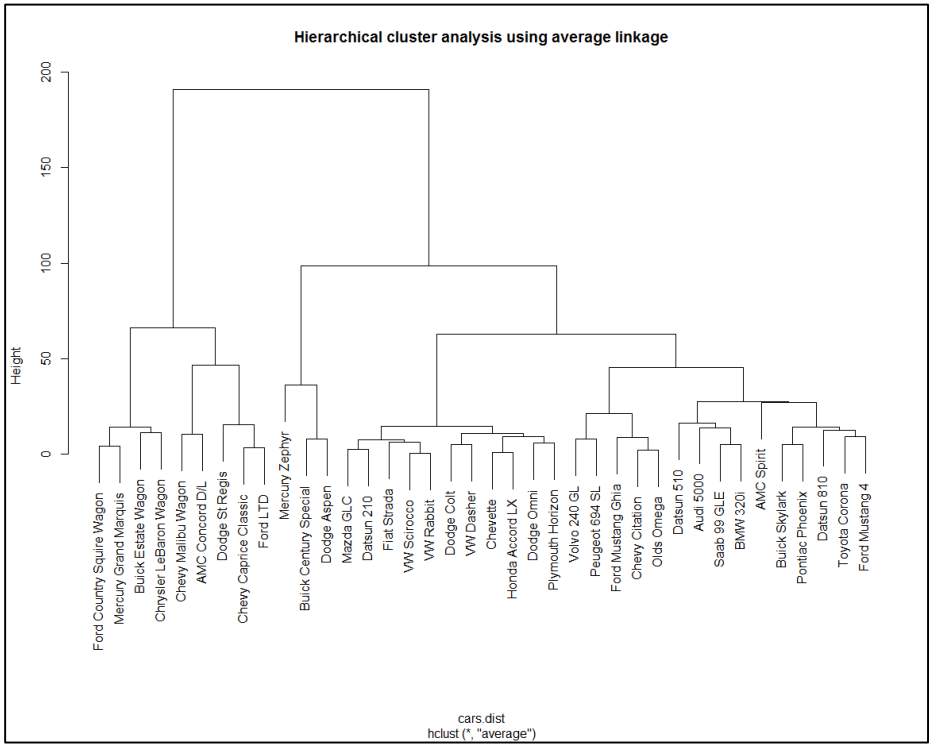
Step 2: Partitioning (if k-means)

```
# K-Means Cluster Analysis
> fit <- kmeans(cars.data, 5) # 5 cluster solution
> fit
K-means clustering with 5 clusters of sizes 10, 4, 6, 11, 7

Cluster means:
       MPG   Weight Drive_Ratio Horsepower Displacement Cylinders
1 25.59000 2.638100    3.298000   96.50000    133.50000       4.3
2 19.12500 3.503750    2.682500  115.00000    245.25000       6.5
3 21.91667 2.970833    3.128333  113.66667    173.83333       6.0
4 32.43636 2.078636    3.477273   70.90909     94.63636       4.0
5 17.17143 3.957714    2.402857  139.85714    333.85714       8.0

Clustering vector:
       ...

# get cluster means
aggregate(cars.data, by=list(fit$cluster),FUN=mean)

# append cluster assignment
cars.data<- data.frame(cars.data, fit$cluster)
```

Reference: http://www.statmethods.net/advstats/cluster.html

Step 3: Visualizing

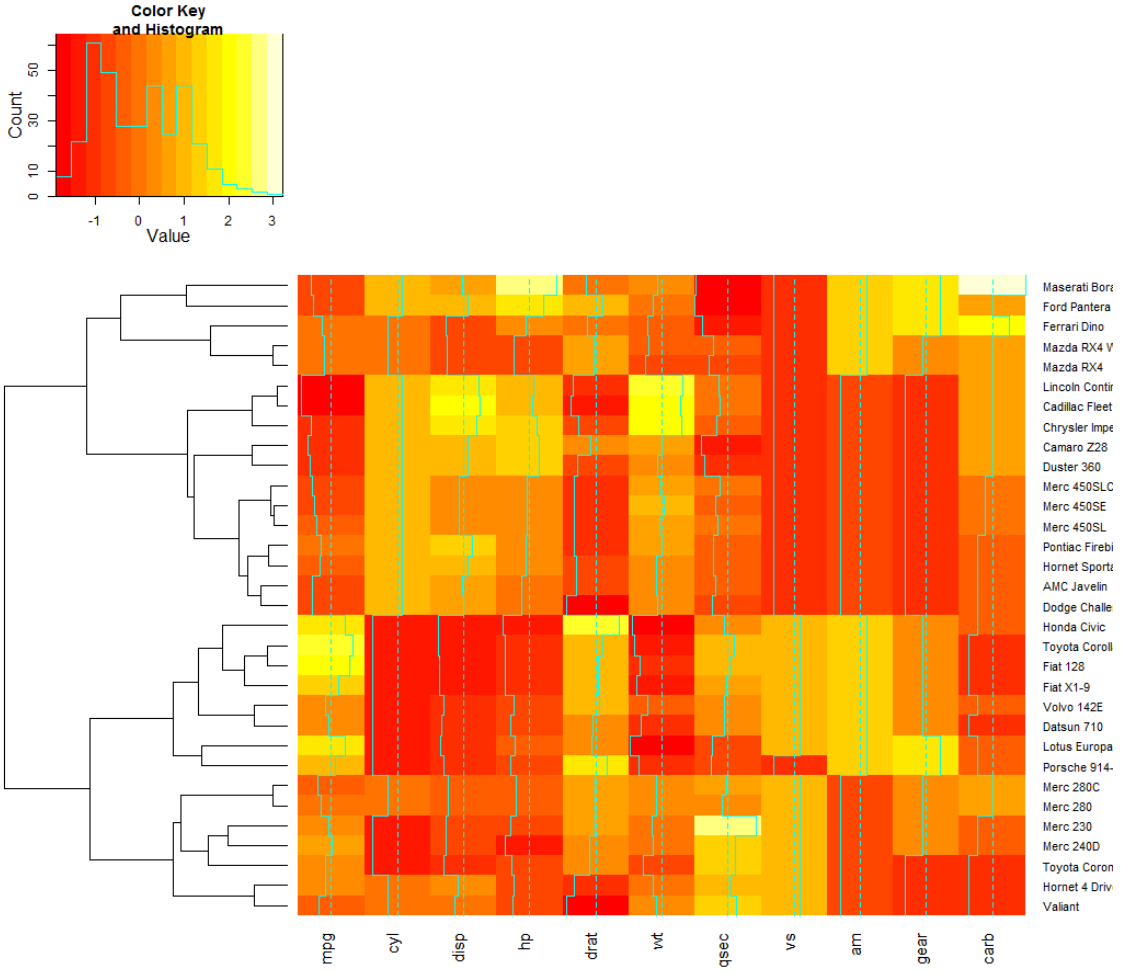
```
plot(cars.hclust,labels=cars$Car,main='Hierarchical cluster analysis using average linkage')
```



Reference: http://www.statmethods.net/advstats/cluster.html

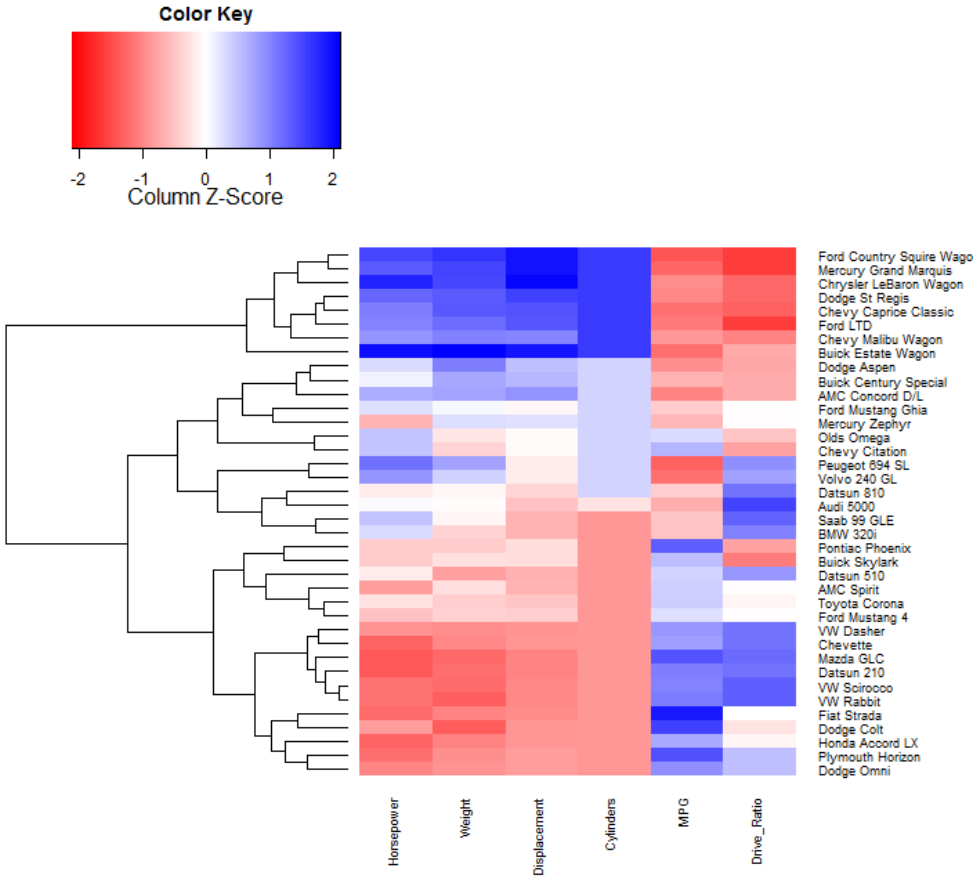
New clustering in R: heatmap.2

```
> library(gplots)
> mtscaled <-
as.matrix(scale(mtcars))
> heatmap.2(mtscaled,
Colv=F, scale='none')
```



Clustering in R: heatmap.2

```
install.packages("gplots")
library(gplots)
heatmap.2(as.matrix(cars.data),
hclustfun = function(x)
    hclust(x,method = "average"),
scale = "column",
dendrogram="row",
trace="none",
density.info="none",
col=redblue(256),
lhei=c(2,5.0), lwid=c(1.5,2.5),
keysize = 0.25,
margins = c(5, 8),
cexRow=0.7,cexCol=0.7)
```



How many clusters?

- Theoretical, conceptual or practical issues may suggest a certain number of clusters
- Hierarchical clustering:
  - Distance threshold at which clusters are combined
- K-means and other non-hierarchical
  - Ratio of total within-groups variance to between-group variance, vs # of clusters
  - Within-groups sum of squares vs # of clusters
  - Plot #clusters vs. total within sum of squares
    - Elbow/sharp bend shows point at which adding more clusters helps reduce distortion measure less and less

How many clusters?

- Theoretical, conceptual or practical issues may suggest a certain number of clusters
- Hierarchical clustering:
  - Distance threshold at which clusters are combined
- K-means and other non-hierarchical
  - Elbow method
  - Silhoutte method

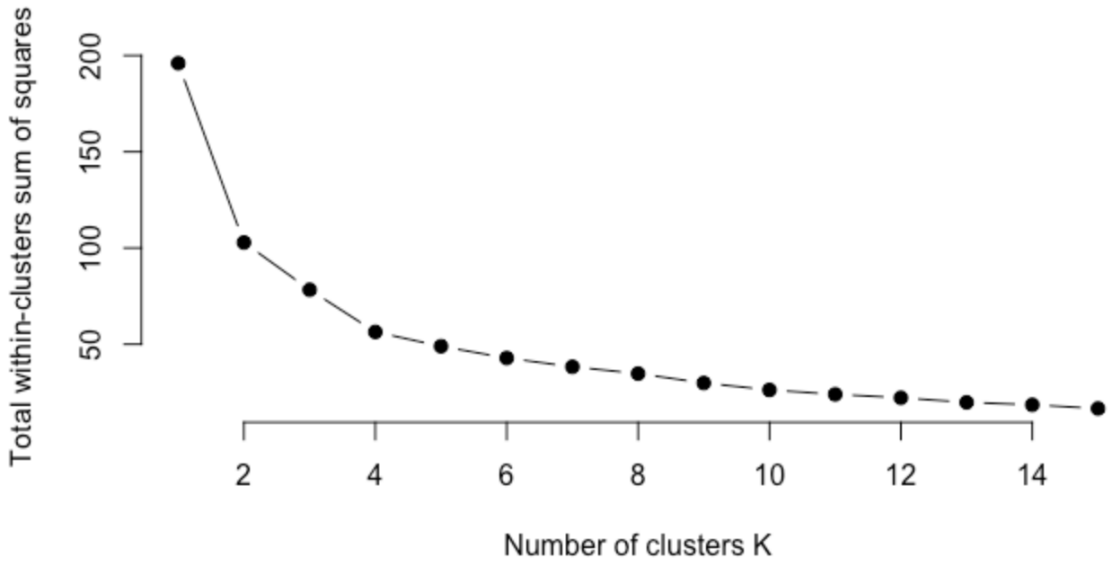
Elbow method

- We can use the total intra-cluster variation (known as total within-cluster variation or total within-cluster sum of square) to quantify how good a clustering technique is doing in identifying clusters.
- This is also the percentage of variance explained by the use of the clusters.
- How can we use this to choose the right k? How about:

$$minimize\left(\sum_{i=1}^{k} W(C_i)\right)$$

where W(C_i) is the within-cl

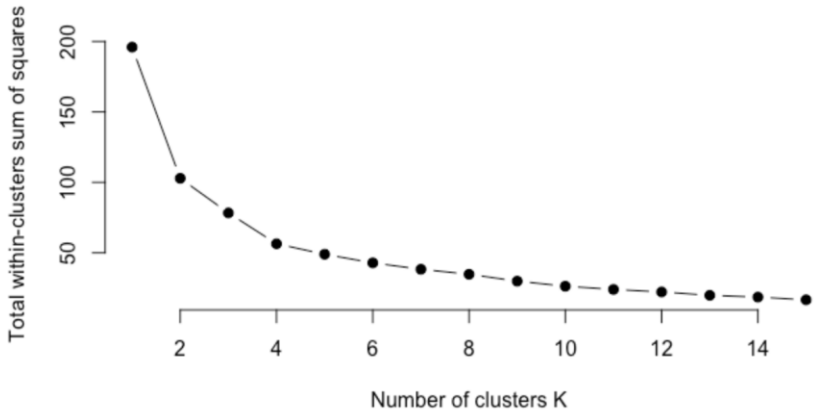
- Plot #clusters vs. tot
  - Elbow/sharp ber
    distortion measu

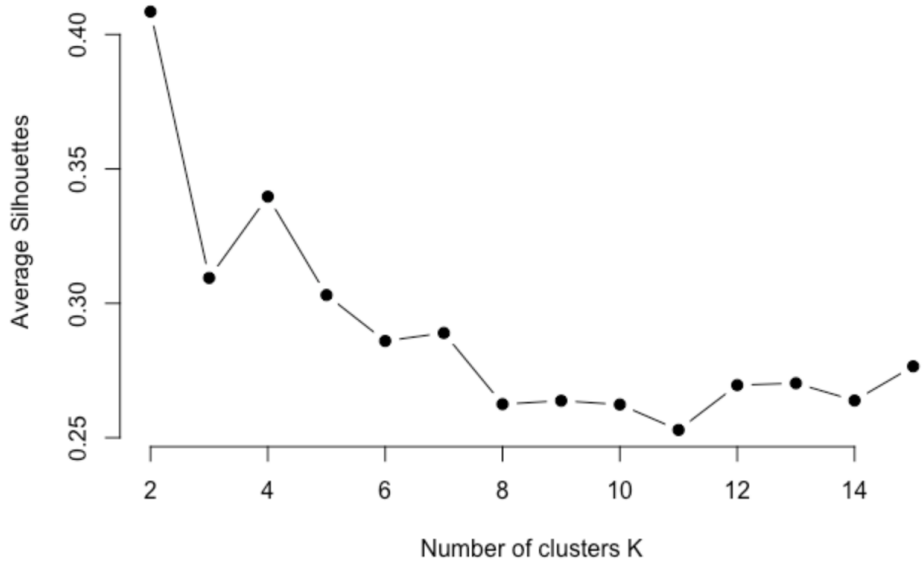


Silhouette Method

- Elbow only looks at within cluster quality. Using the silhouette method, you can define quality w.r.t. how well points match their assigned cluster (cohesion) compared to other clusters (separation).
- Fit to current cluster for data point i: $a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$
- Fit to other clusters: $b(i) = \min_{k \neq i} \frac{1}{|} \sum d(i, j)$

$b(i) - a(i$



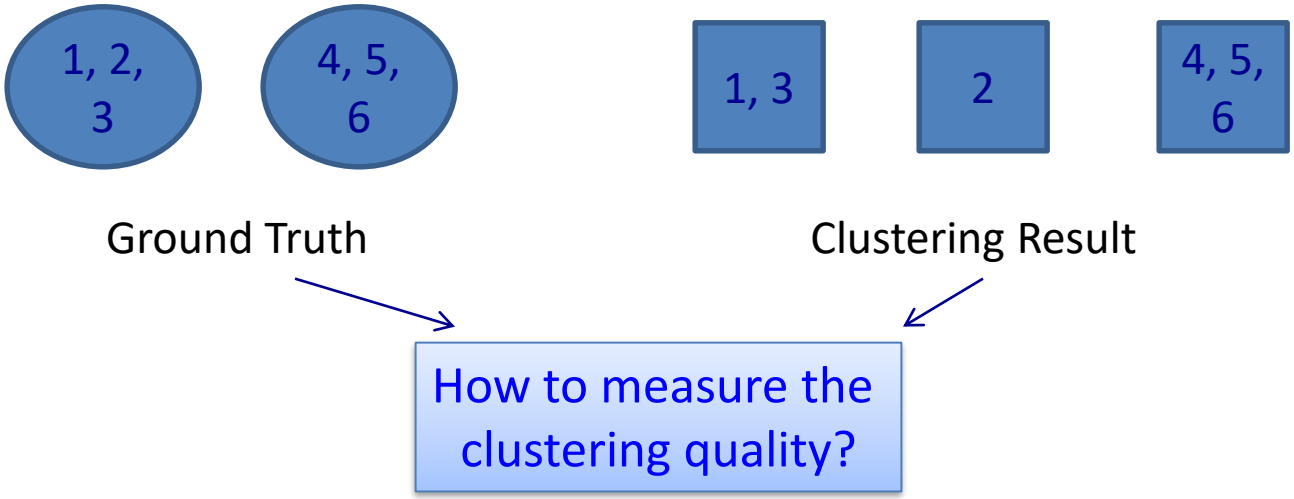


Elbow and silhouette methods do not always give the same answer!

How do we know if we've found good quality clusters?

- Compare cluster stability across:
  - Different distance measures
  - Different clustering methods
  - Different 50/50 random data splits
  - Different variable/features deletions
  - Different data orderings (non-hierarchical)
- "Good" clusterings (if they exist) are generally stable and robust to perturbations in methods or data

Measuring a Clustering Result: If you know the ground truth



- The number of clusters after grouping can be different from the ground truth
- No clear correspondence between clustering result and the ground truth

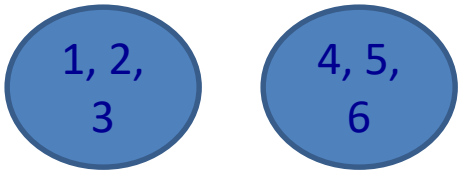
Accuracy of Pairwise Cluster Memberships

- Consider all the possible pairs of nodes and check whether they reside in the same cluster
- An error occurs *if*
  - Two elements belonging to the same true cluster are assigned to different clusters after clustering
  - Two nodes belonging to different true clusters are assigned to the same cluster
- Construct a contingency table

		Ground Truth	
		$C(v_i) = C(v_j)$	$C(v_i) \neq C(v_j)$
Clustering Result	$C(v_i) = C(v_j)$	a	b
	$C(v_i) \neq C(v_j)$	c	d

$$accuracy = \frac{a+d}{a+b+c+d} = \frac{a+d}{n(n-1)/2}$$

Accuracy Example



Ground Truth

Clustering Result

		Ground Truth	
		$C(v_i) = C(v_j)$	$C(v_i) != C(v_j)$
Clustering Result	$C(v_i) = C(v_j)$	4	0
	$C(v_i) != C(v_j)$	2	9

Accuracy = (4+9)/ (4+2+9+0) = 13/15

What you should know

- Basic use of hierarchical and k-means clustering
- When is k-means clustering preferable to hierarchical clustering? Or vice-versa?
- How is cluster quality measured?
- How to do clustering in R