SI 618
Exploratory Data Analysis

Clustering Analysis

Instructor: Dr. Chris Teplovs (cteplovs@umich.edu)

Lead Developer, Digital Innovation Greenhouse, Office of Academic Innovation

Adjunct Lecturer of Information, School of Information

GSI: SungJin Nam (sjnam@umich.edu)

Course announcements

- This week:
  - Homework 3 due today
  - Homework 4 (Cluster Analysis) released

- No class next week (Thanksgiving)

      » AND……….

This course has been shortened!

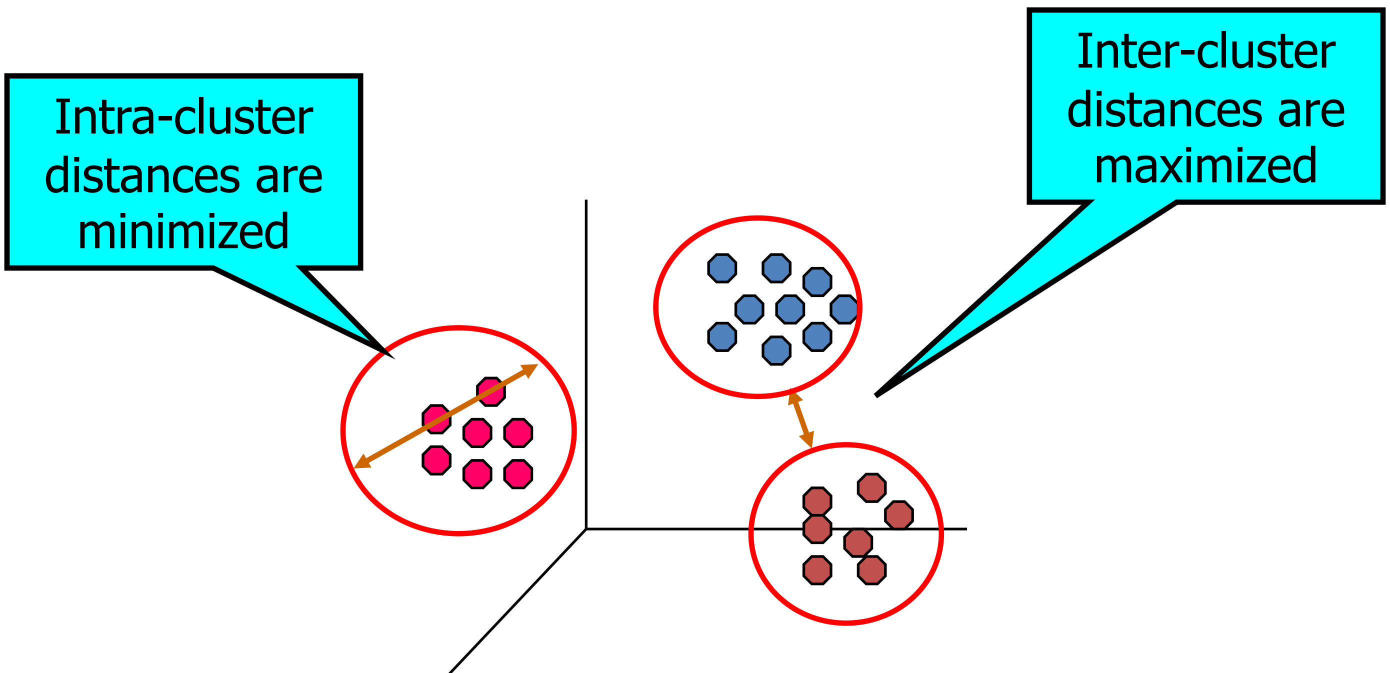
- Last day of class is DECEMBER 9, 2016
  - Per information from the SI Registrar
- Due date for project is still DECEMBER 16, 2016
- SLIDES ASSIGNMENT HAS BEEN ELIMINATED
- December 9 class will be brief intro to machine learning, review of 618, and teaching evaluations

SI 618 Data Exploration: Class Schedule

Date	Topic	Assignments Due
Week 1	Course introduction Basics of Programming with R	
Week 2	Basic analysis and visualization using ggplot2: qplot() Manipulating data frames using plyr	Homework 1
Week 3	Smoothing and Trend-finding. Building ggplot Layer by Layer	Homework 2
Week 4	Cluster analysis	Homework 3
Week 5	(Thanksgiving: no class!)	
Week 6	Factor Analysis Methods (PCA, EFA)	Homework 4
Week 7	Machine Learning, Review, Evaluations	

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



Applications of Cluster Analysis

- **Understanding**
  - Group related documents for browsing
  - Group genes and proteins that have similar functionality
  - Group stocks with similar price fluctuations

	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP

- **Summarization**
  - Reduce size of large data sets

Clustering precipitation in Australia

Summary: Conducting Cluster Analysis

Formulate the Problem

↓

Select a Distance Measure

↓

Select a Clustering Procedure

↓

Decide on the Number of Clusters

↓

Interpret and Profile Clusters

↓

Assess the Validity of Clustering

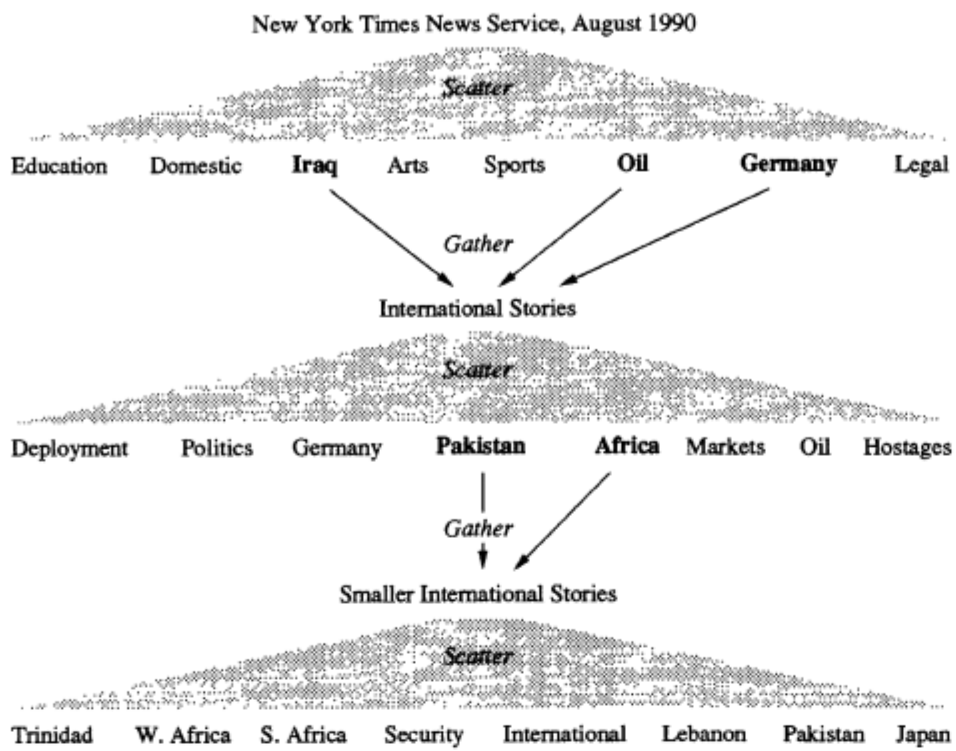
Clustering is often used as an exploratory data analysis tool

- 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

Example: Scatter/Gather. A clustering-based approach to browsing large document collections [Cutting et al. SIGIR 1992]

- What if you have a vague information need that spans topics
- And you're not sure which search terms to use?

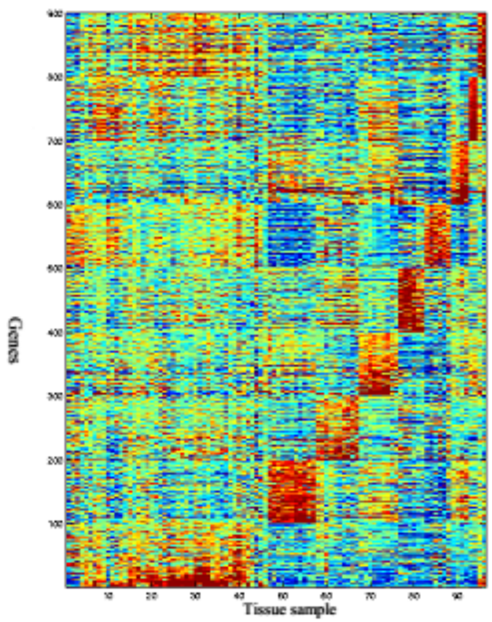
1. Scatter: Present the user with a set of clusters
2. Gather: User selects subset of clusters that seem relevant
3. Combine clusters Repeat from step 1 until done.

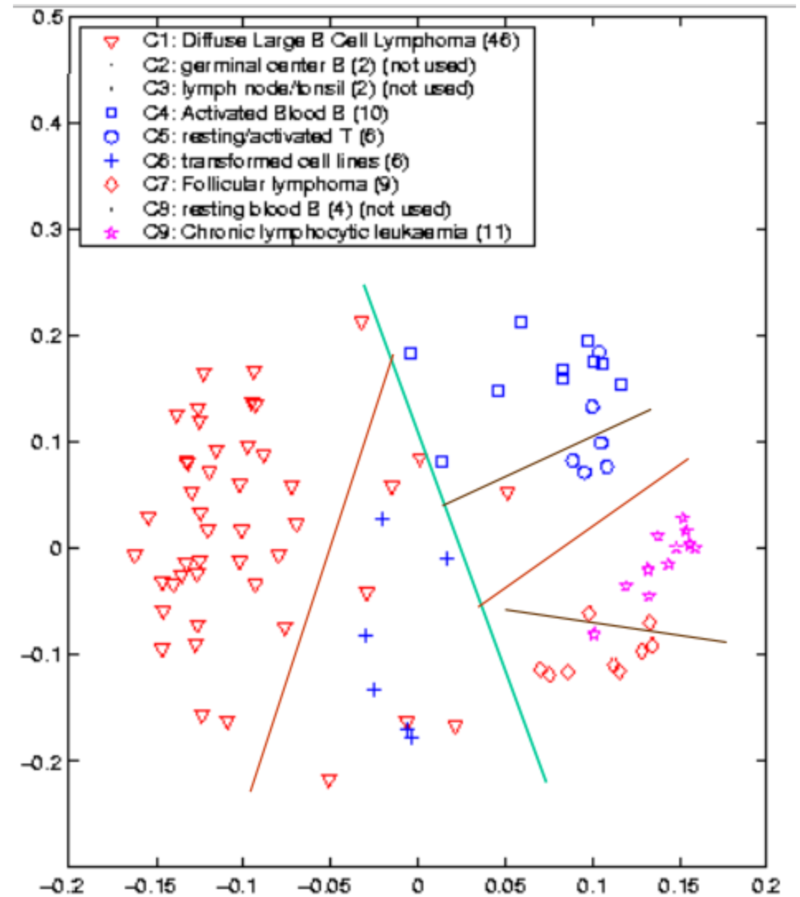


Source http://dl.acm.org/citation.cfm?id=133214

Example: Clustering lymphoma cancer tissue samples

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



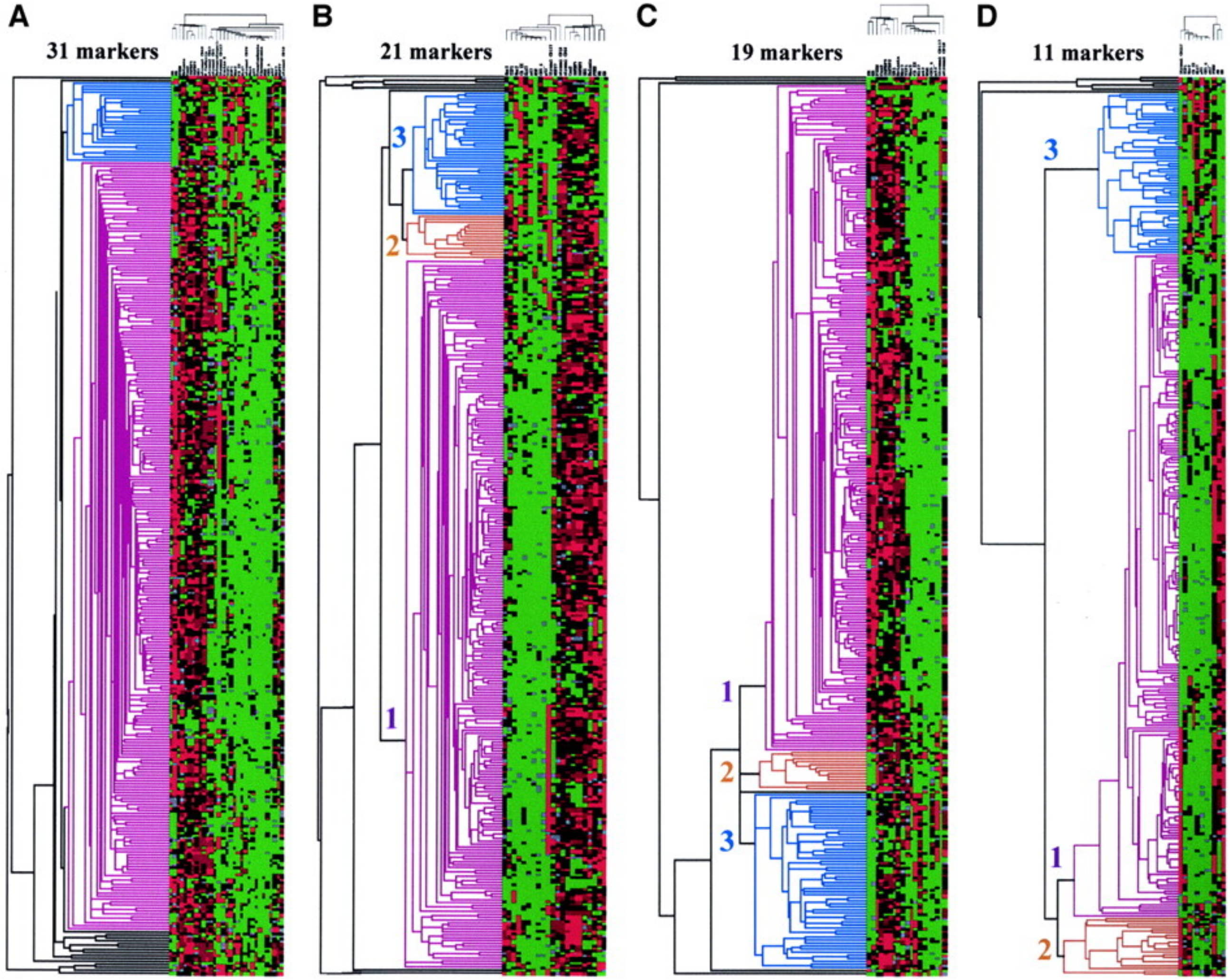


[Source: Chris Ding, ICML 2004 Tutorial on Spectral Clustering]

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)

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

Clinical Cancer Research

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

How many clusters?

Six Clusters

Two Clusters

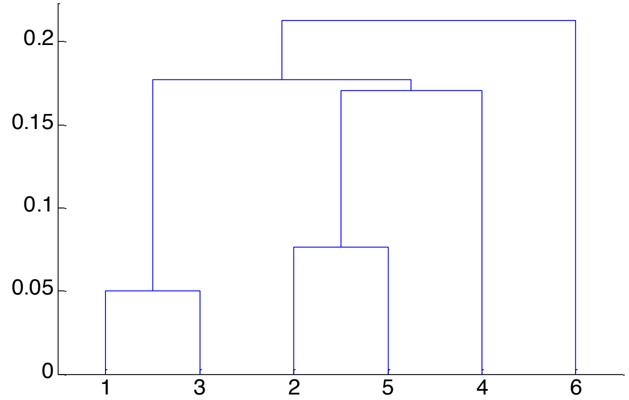
Four Clusters

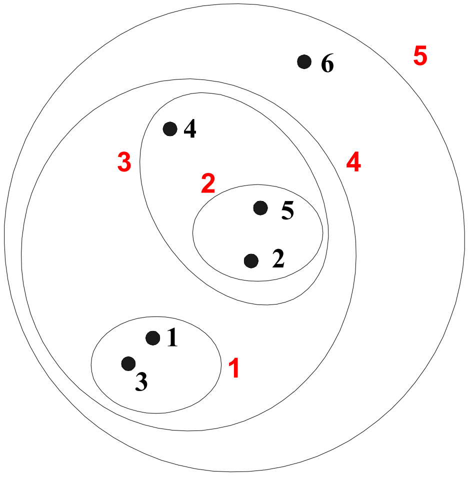
What algorithms are used to find clusters? Answer: huge range of approaches

- 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:
  - **Heirarchical**: agglomerative, single-link, etc.
  - **Partitional**: k-means, k-median, etc.

Hierarchical Clustering

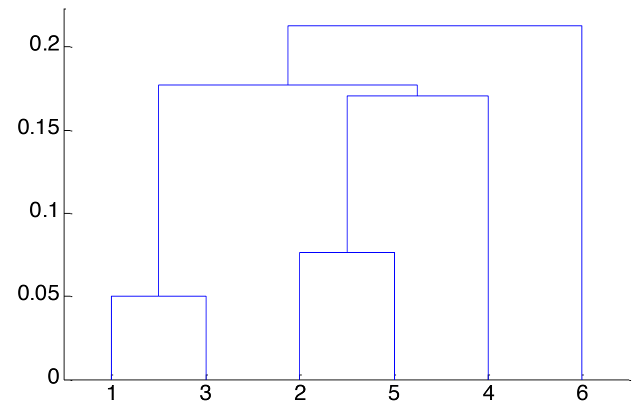
- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





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 dendrogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)



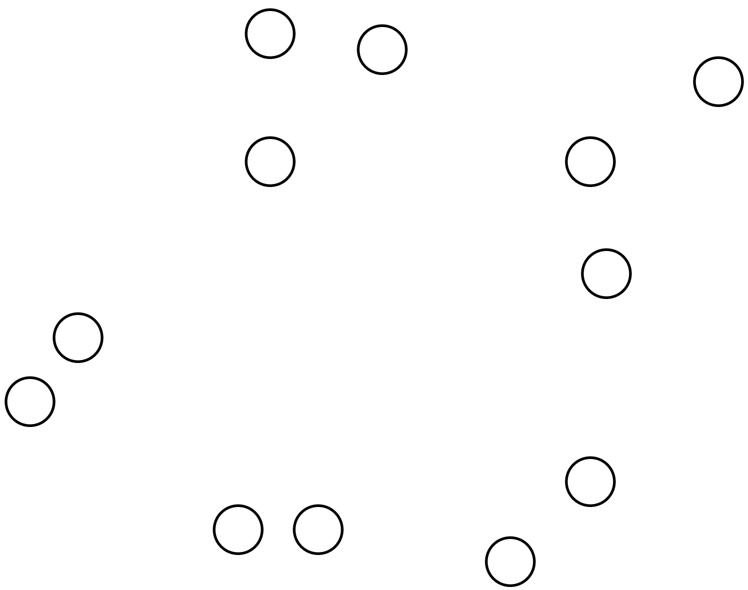
Hierarchical clustering

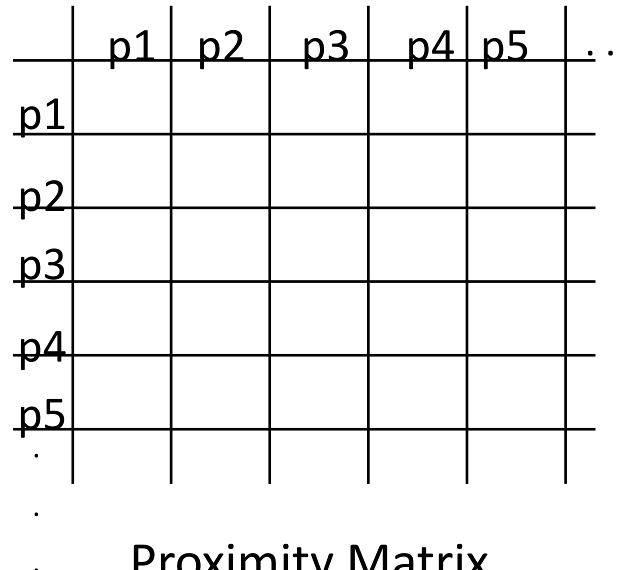
- Bottom-up (‘Agglomerative’)
  - Start with each point being in its own cluster
  - At each step
    - Merge the most similar pair of clusters based on a cost function
    - Continue until you have k clusters, or everything is in one big cluster
- Top-down (‘Divisive’)
  - Start with all points in a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

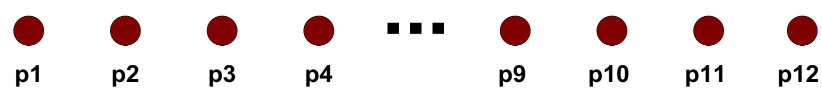
Agglomerative (bottom-up) Clustering: Starting Situation

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





Proximity Matrix

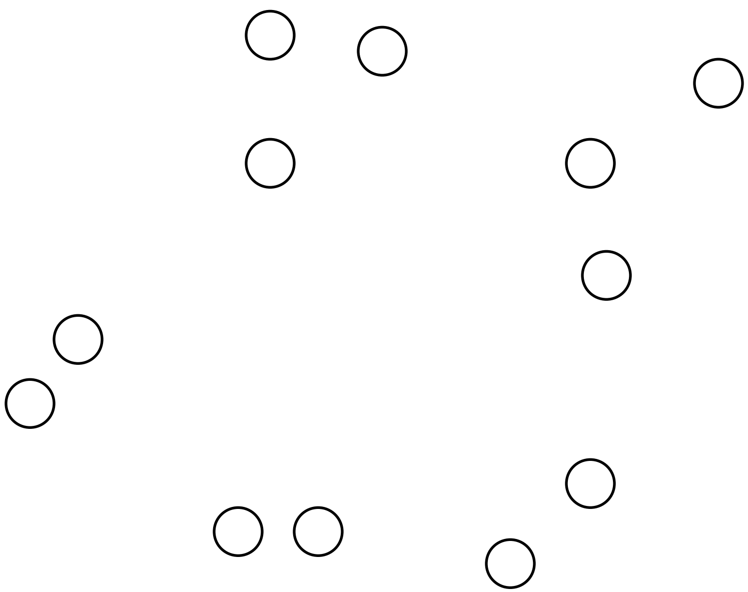


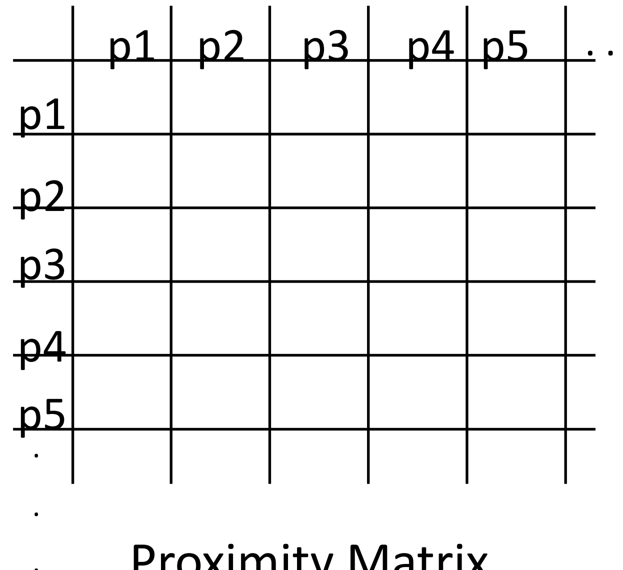
Agglomerative Clustering Algorithm

- 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 <u>cost function</u>.
  – Different approaches to defining the distance between clusters distinguish the different algorithms

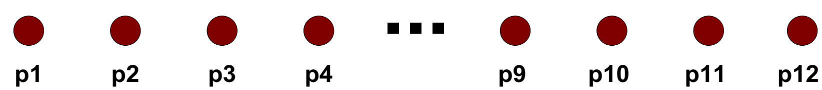
Agglomerative (bottom-up) Clustering: Starting Situation

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



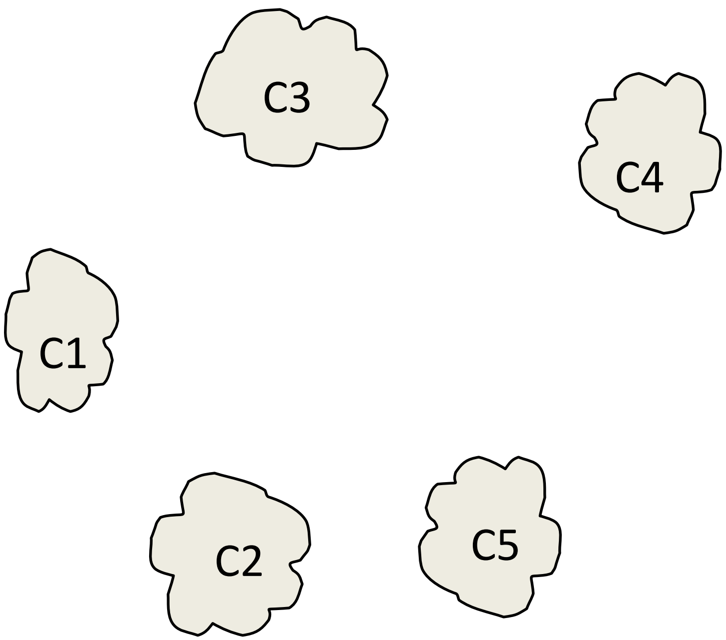


Proximity Matrix



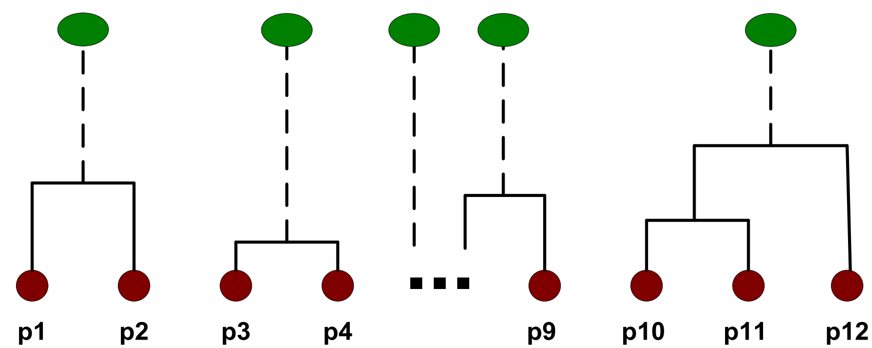
Intermediate Situation

- After some merging steps, we have some clusters



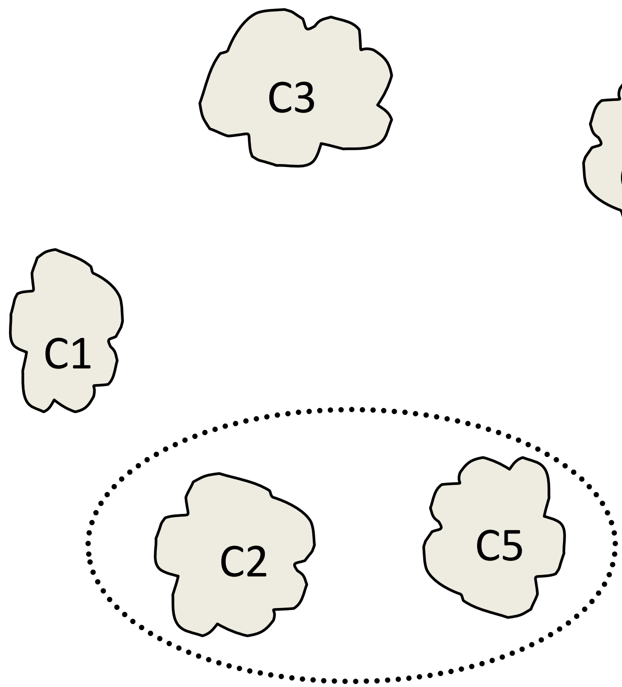
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

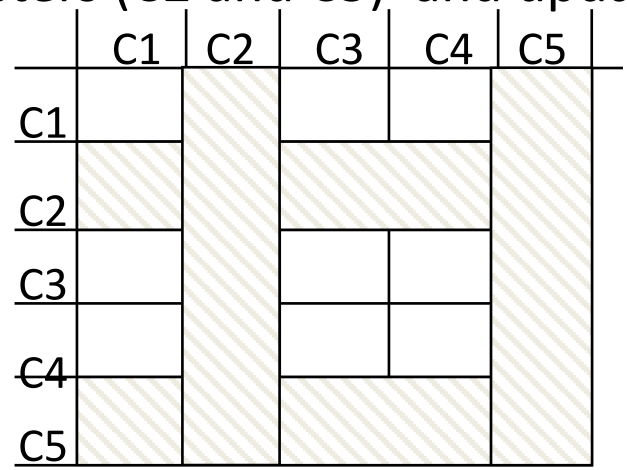
Proximity Matrix



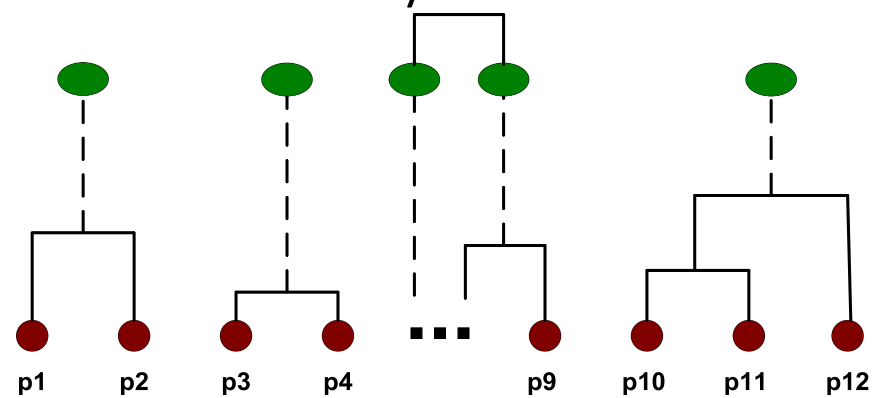
Intermediate Situation

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



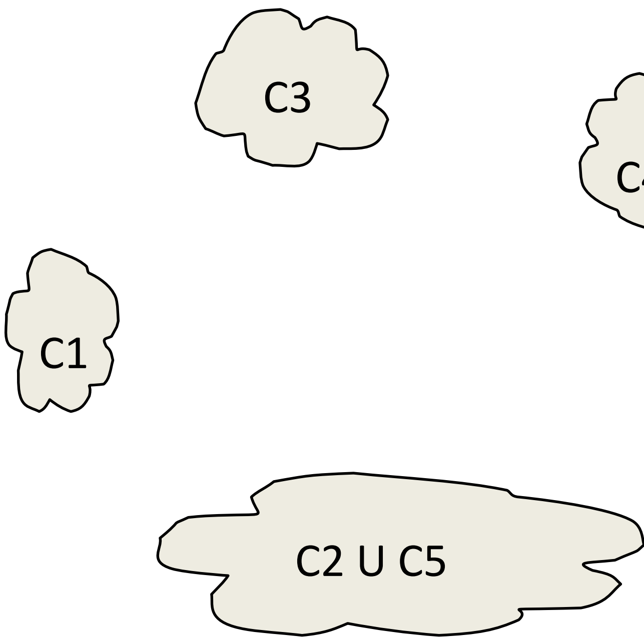


Proximity Matrix



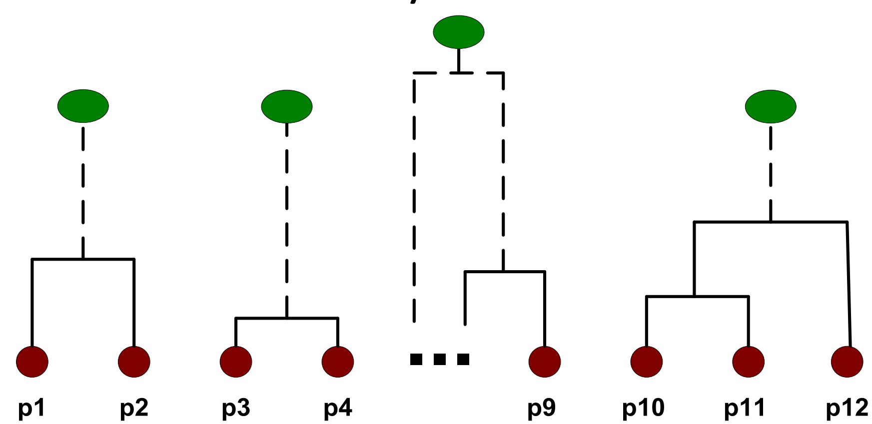
After Merging

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

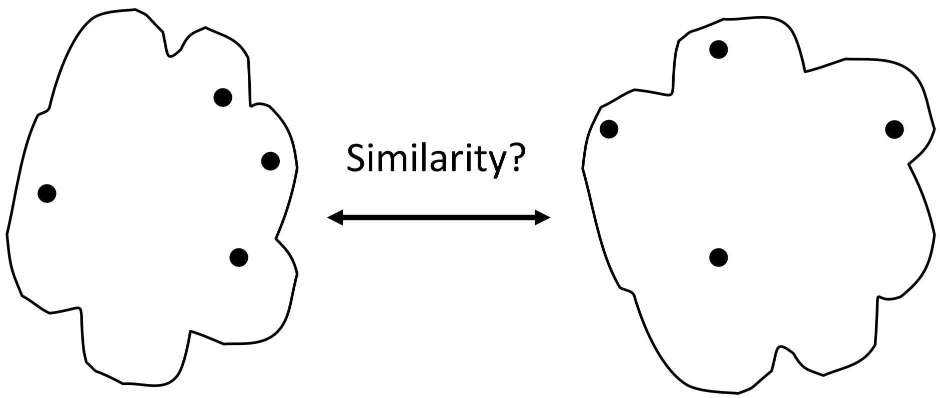


	C1	C2 + C5	C3	C4
C1		?		
C2 + C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

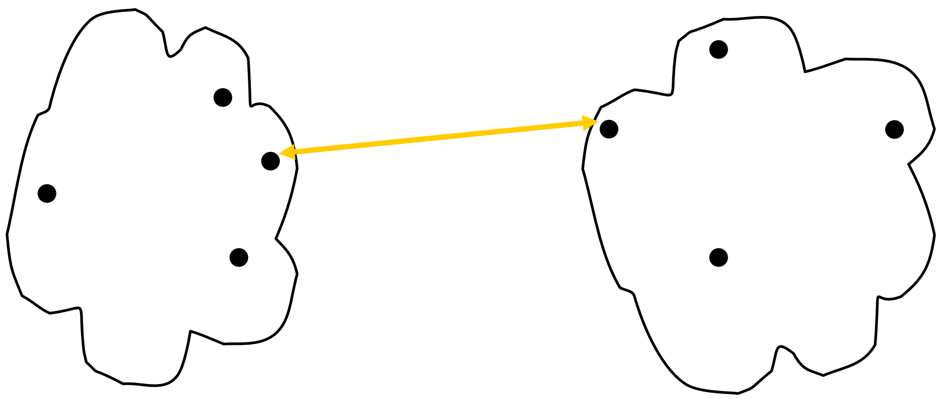


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

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

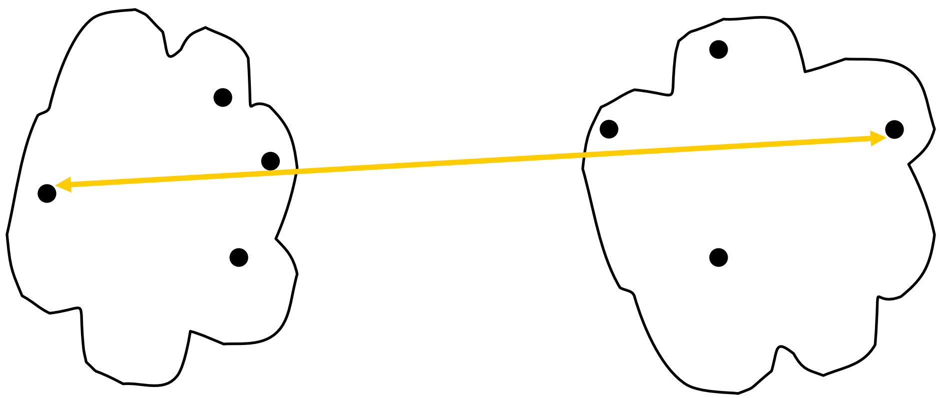


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

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

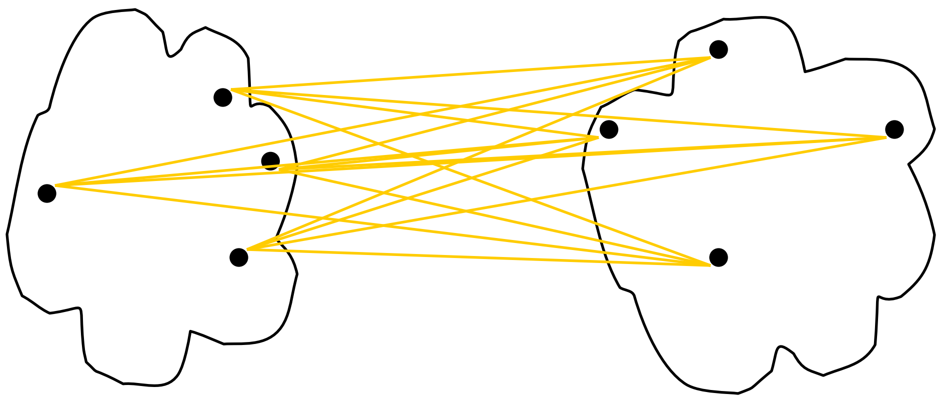


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

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

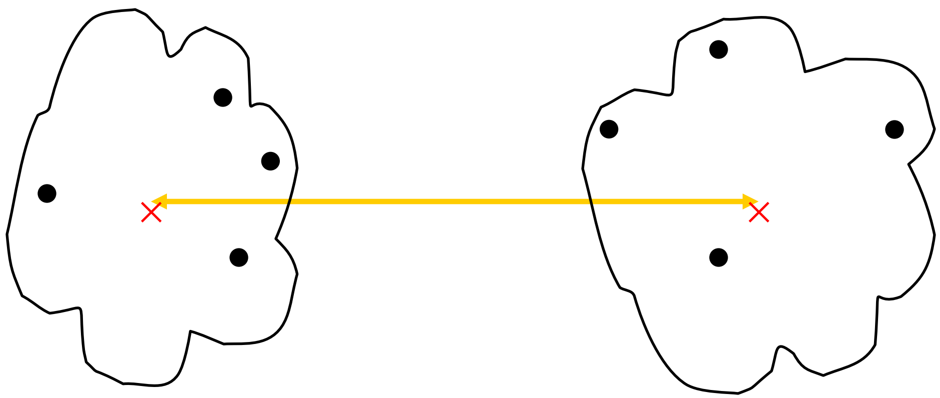


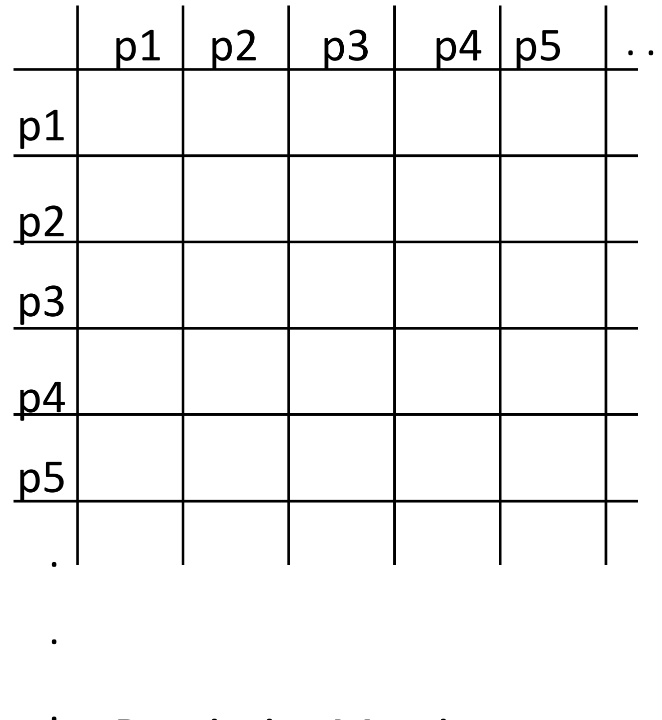
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

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

How to Define Inter-Cluster Similarity



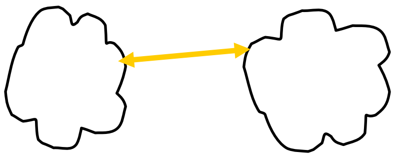


Proximity Matrix

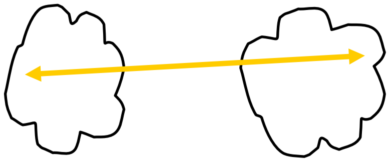
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

Cost functions for bottom-up (agglomerative) clustering

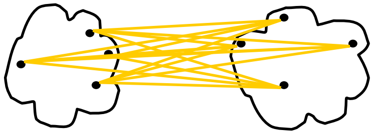
- Single linkage
  - Minimum distance between clusters



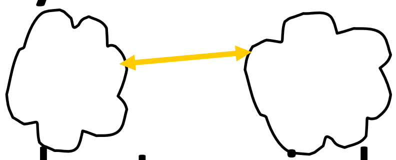
- Complete linkage
  - Max distance between clusters



- Average linkage
  - Average distance between clusters

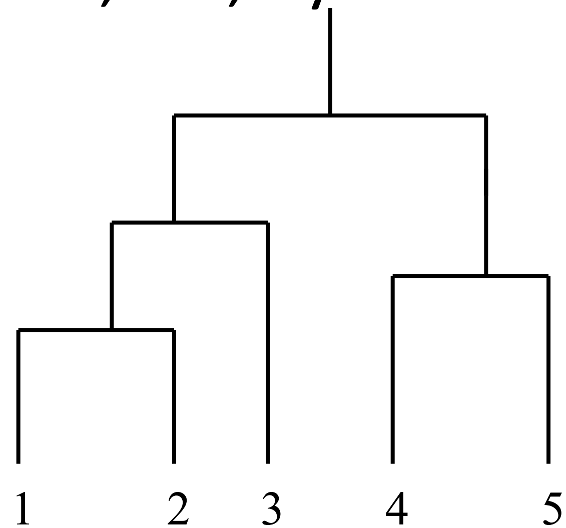


Cluster Similarity: MIN or Single Linkage

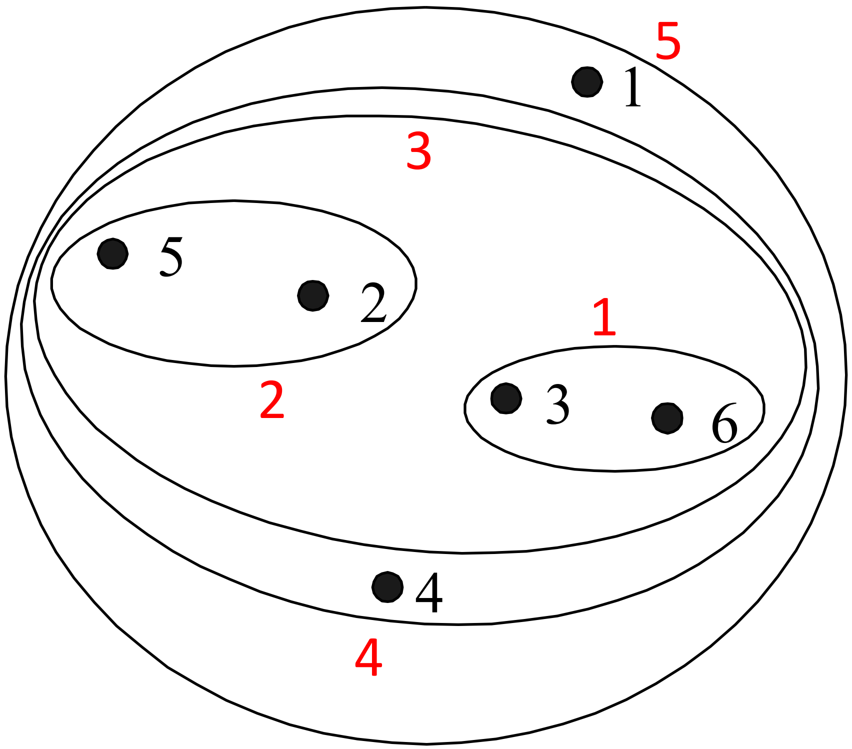


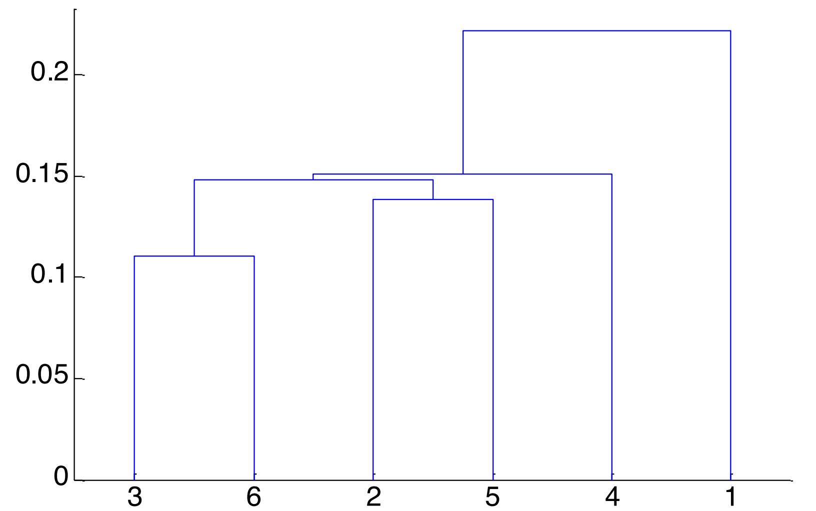
- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

	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

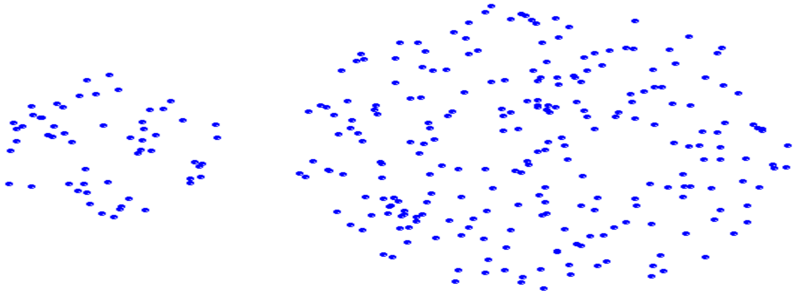




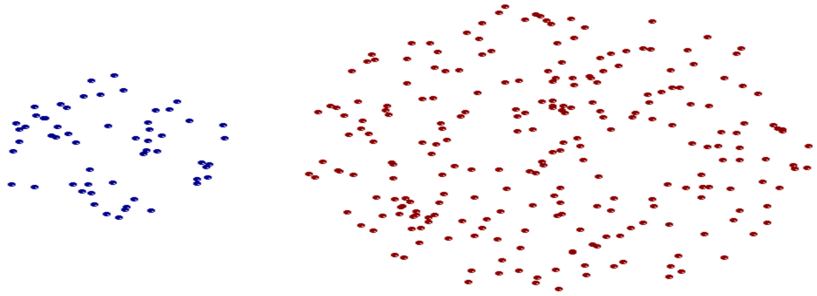
Nested Clusters

Dendrogram

Strength of MIN



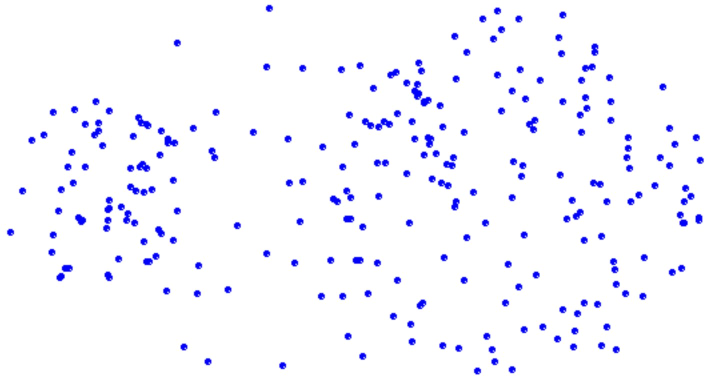
Original Points



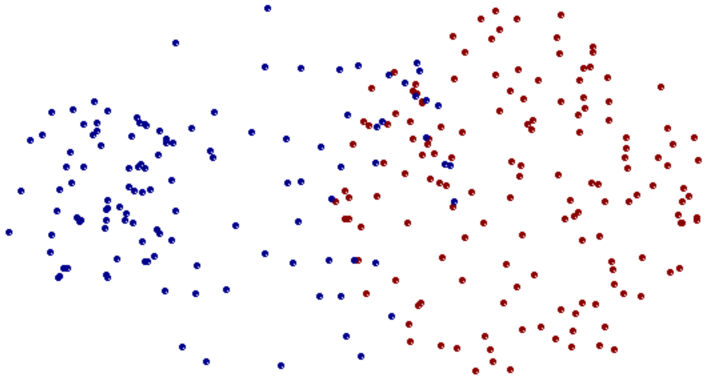
Two Clusters

- Can handle non-elliptical shapes

Limitations of MIN



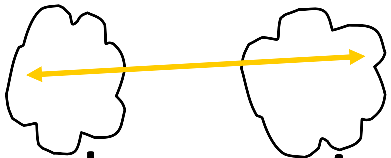
Original Points



Two Clusters

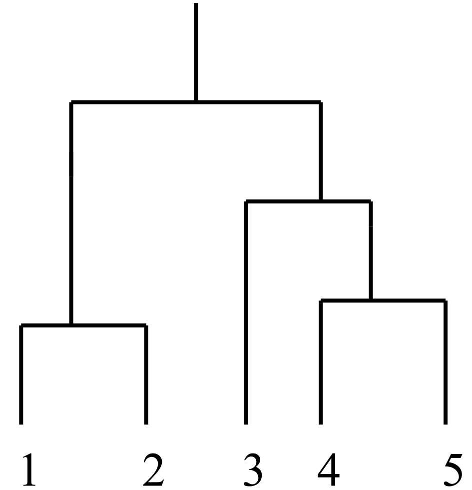
- Sensitive to noise and outliers

Cluster Similarity: MAX or Complete Linkage

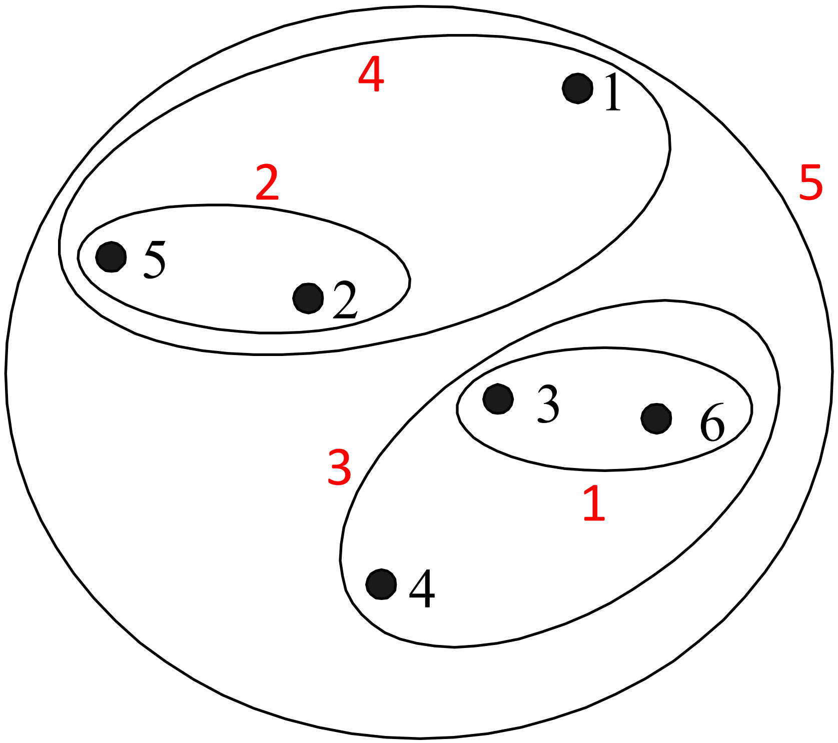


- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

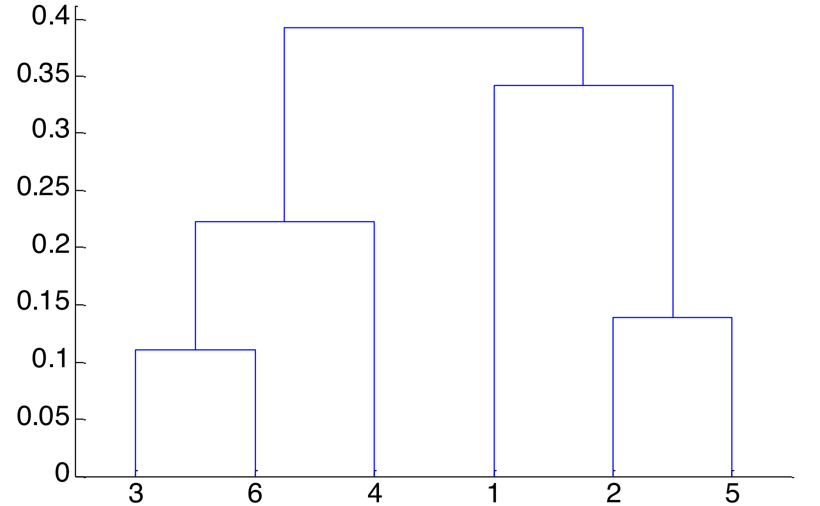
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MAX

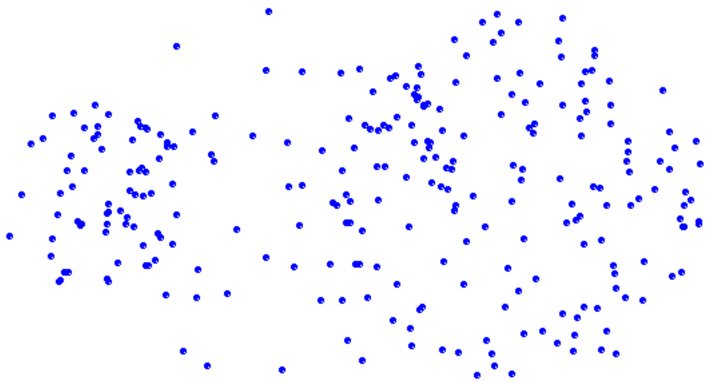


Nested Clusters

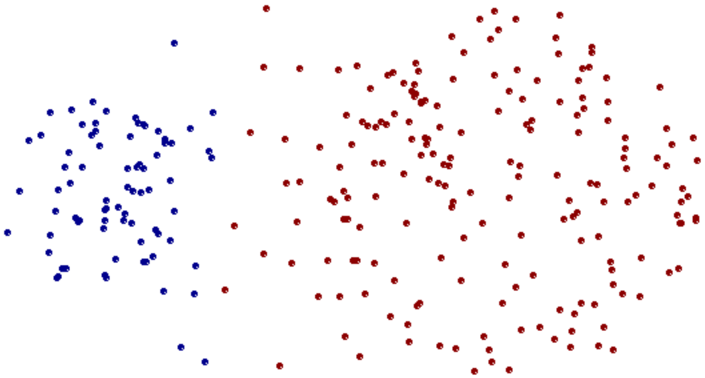


Dendrogram

Strength of MAX



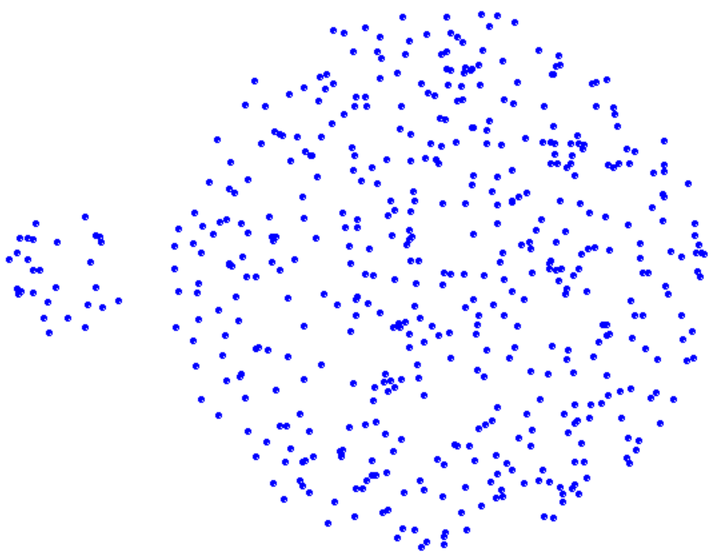
Original Points



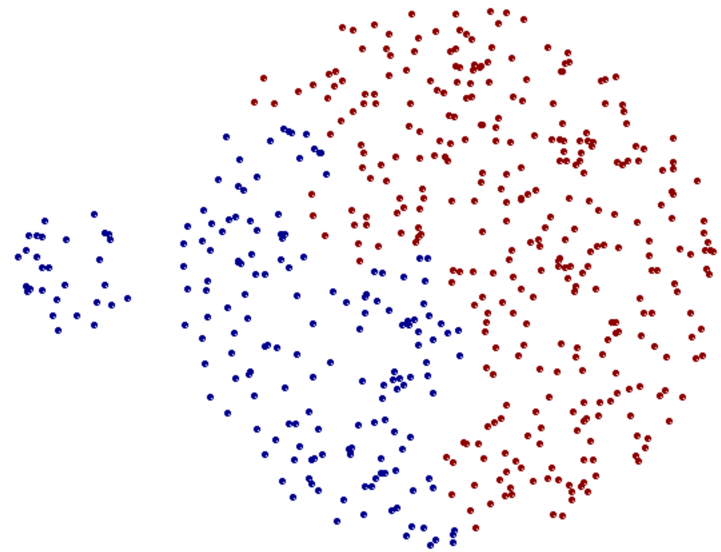
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



Original Points

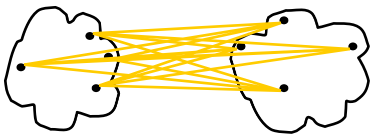


Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

Cluster Similarity: Group Average

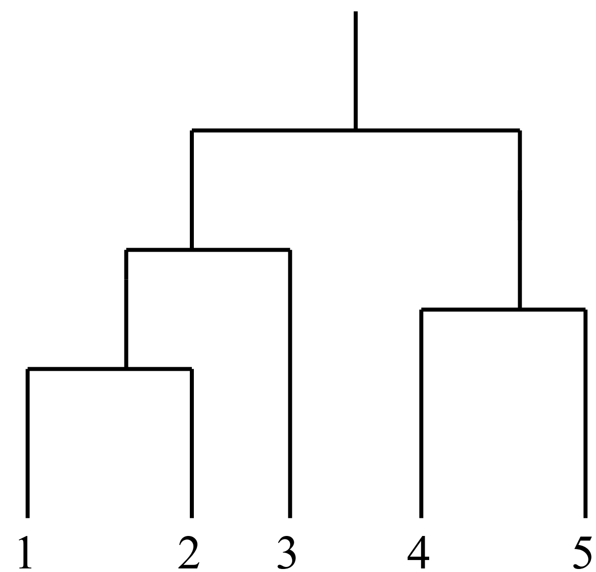
- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.



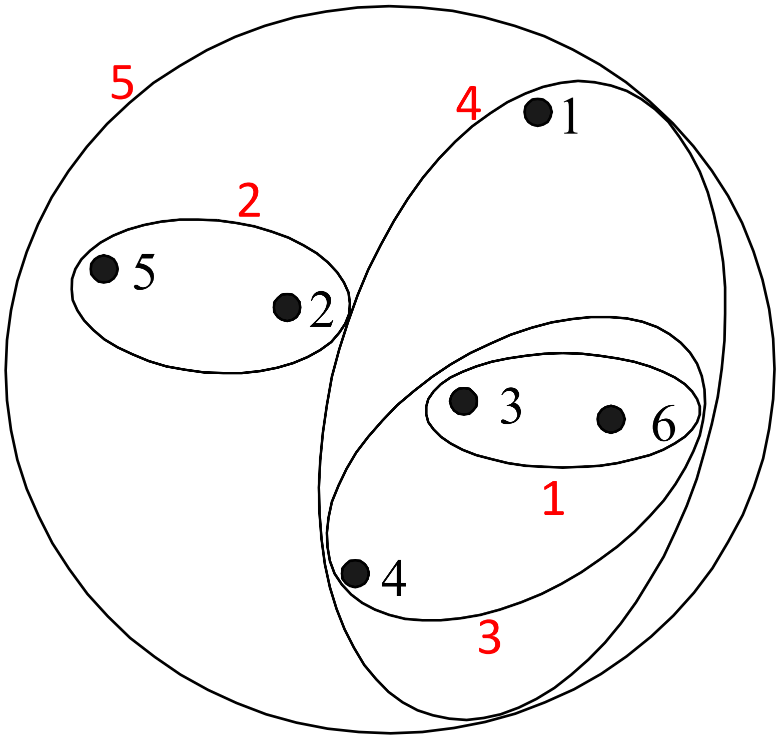
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum\limits_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

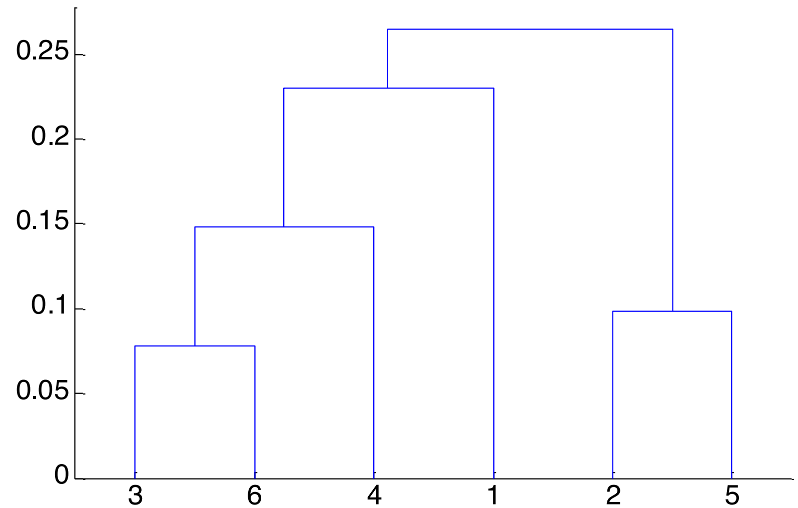
- Need to use average connectivity for scalability since total proximity favors large clusters

	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: 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

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}

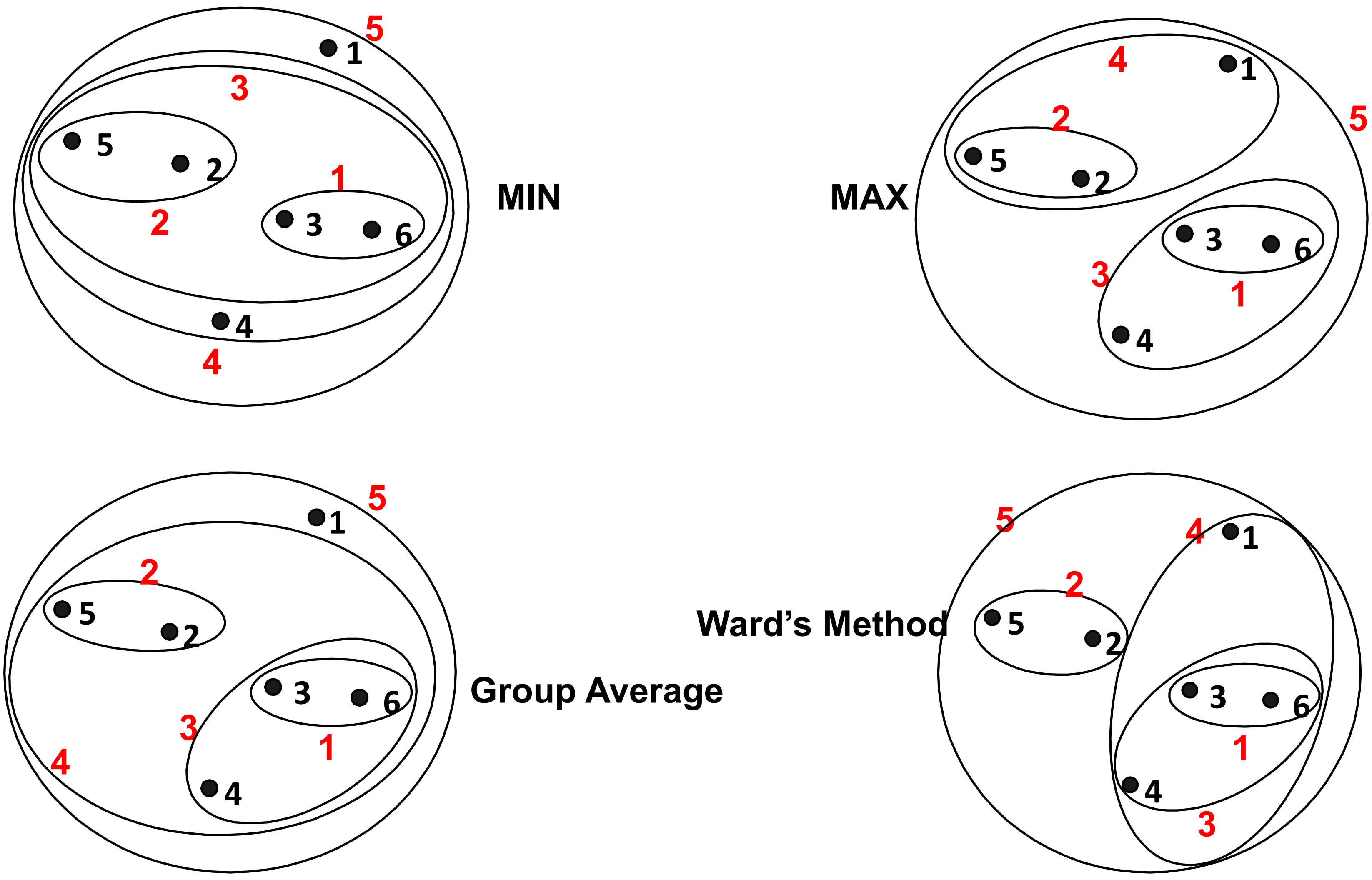
$$D_w(C_i, C_j) = \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 - \sum_{x \in C_{ij}} (x - r_{ij})^2$$

- 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
  - Can be used to initialize k-means

Hierarchical Clustering: Comparison



Which type of hierarchical clustering to use?

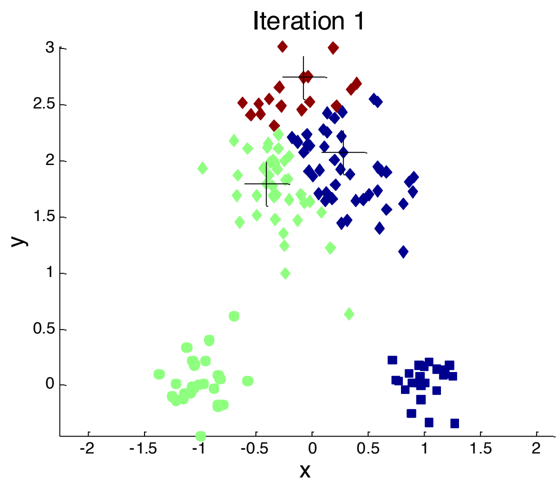
- Different methods have different strengths and weaknesses.
  - Ward's method tends to give equal sized clusters
  - Single linkage (nearest neighbor) tends to make long strings into a cluster.
  - Top-down is sensitive to early errors: bad first choice can wreck the entire process
  - Bottom-up can’t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

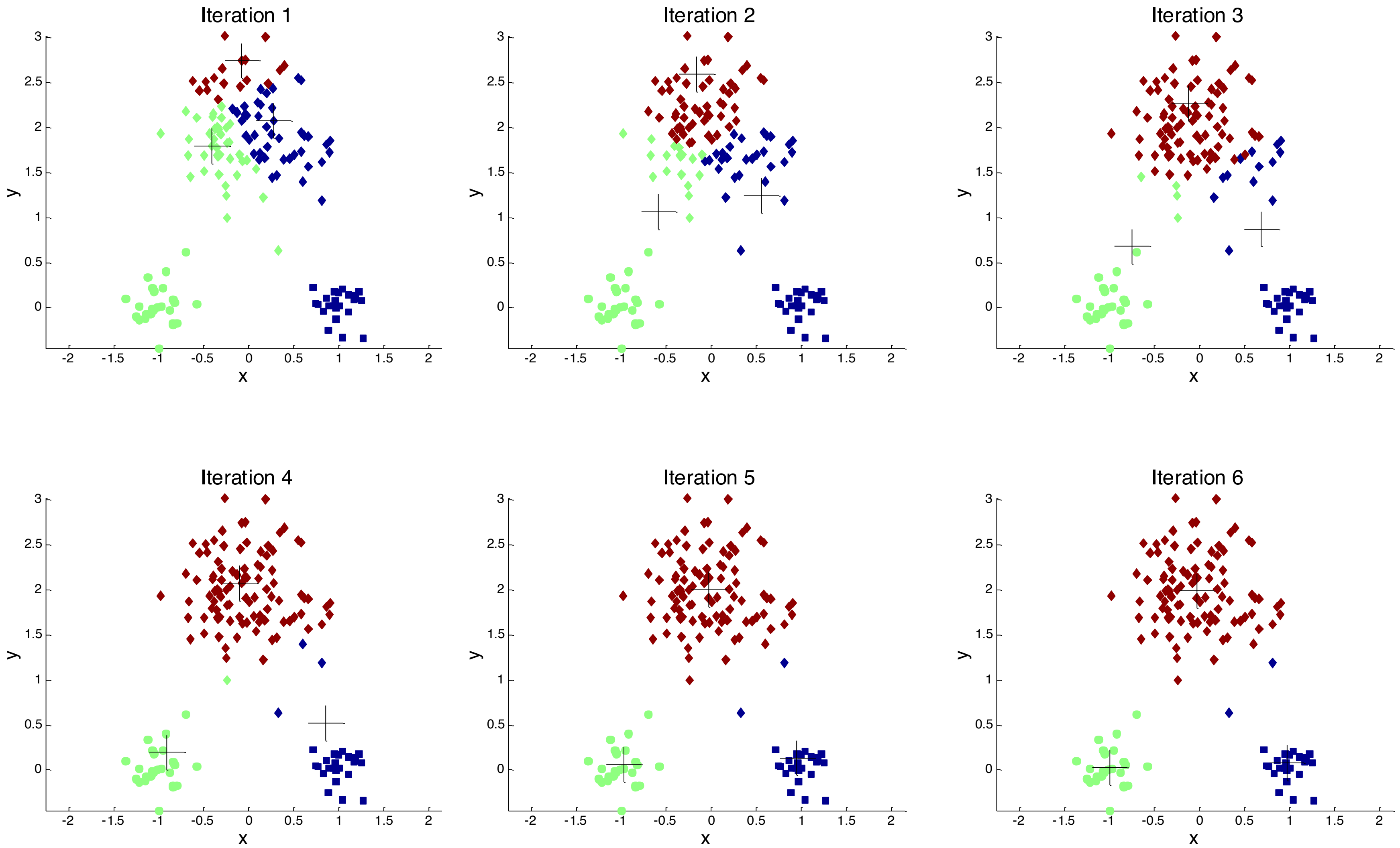
- Partitional clustering approach
- Each cluster associated with a centroid (centerpoint)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified in advance
- The basic algorithm is very simple

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)



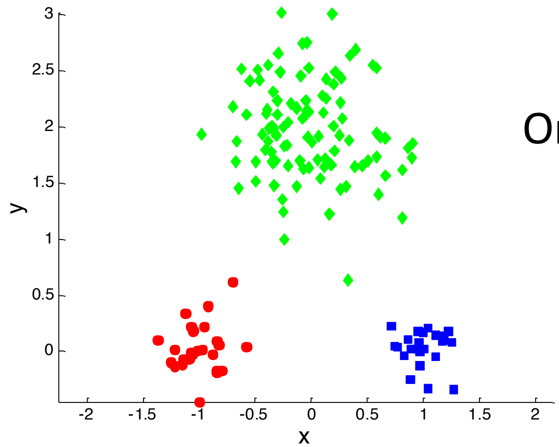
K-means is a special case of model-based clustering

- 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

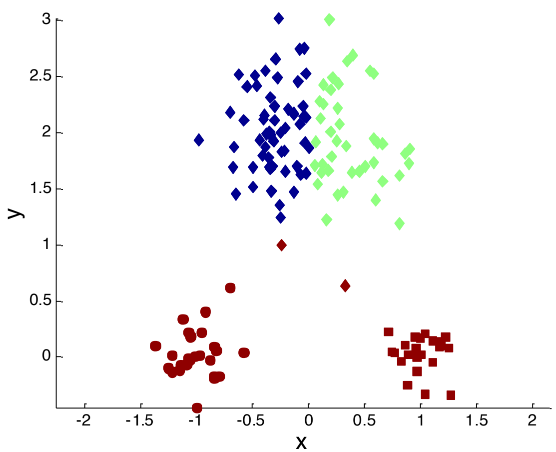
K-means Clustering – Details

- 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

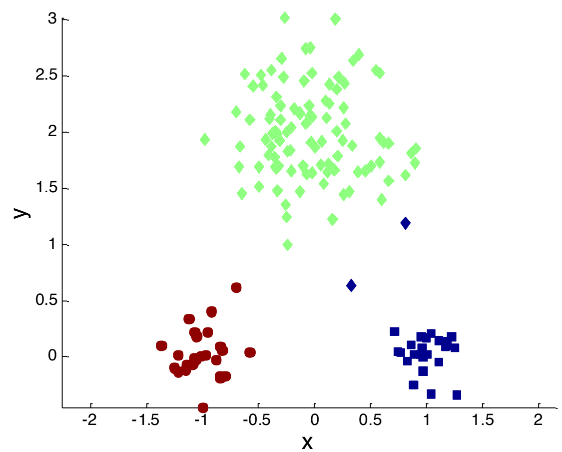
Two different K-means Clusterings



Original Points

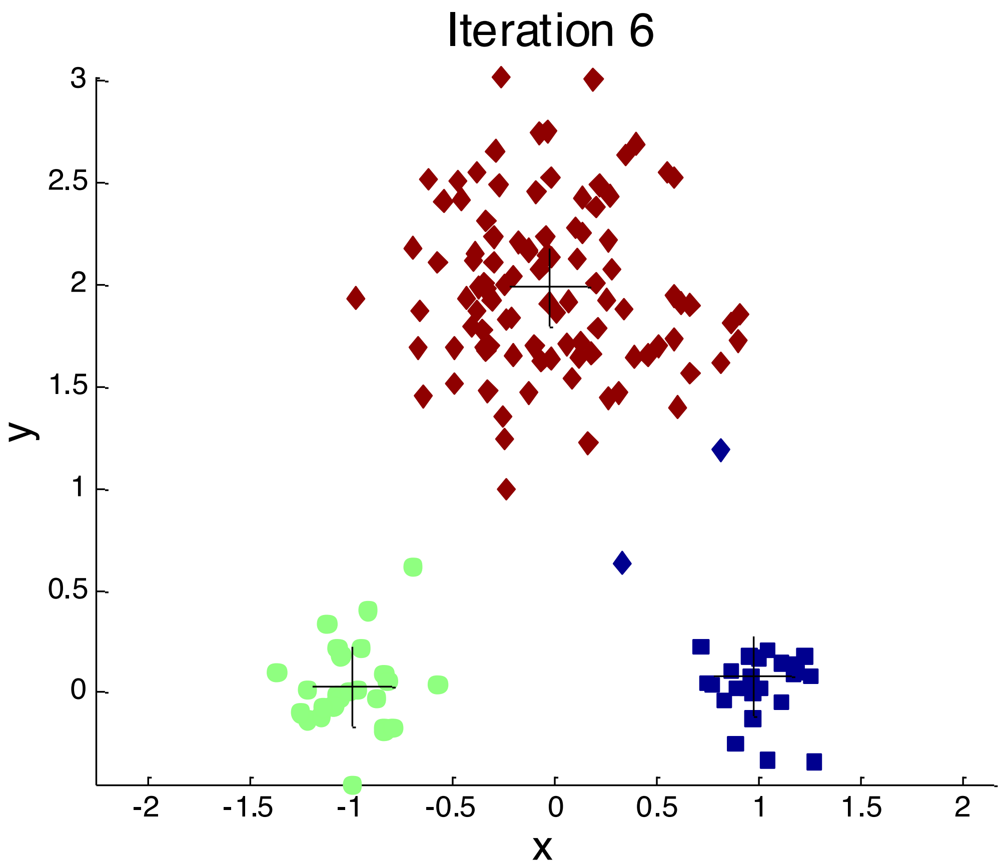


Optimal Clustering? No.

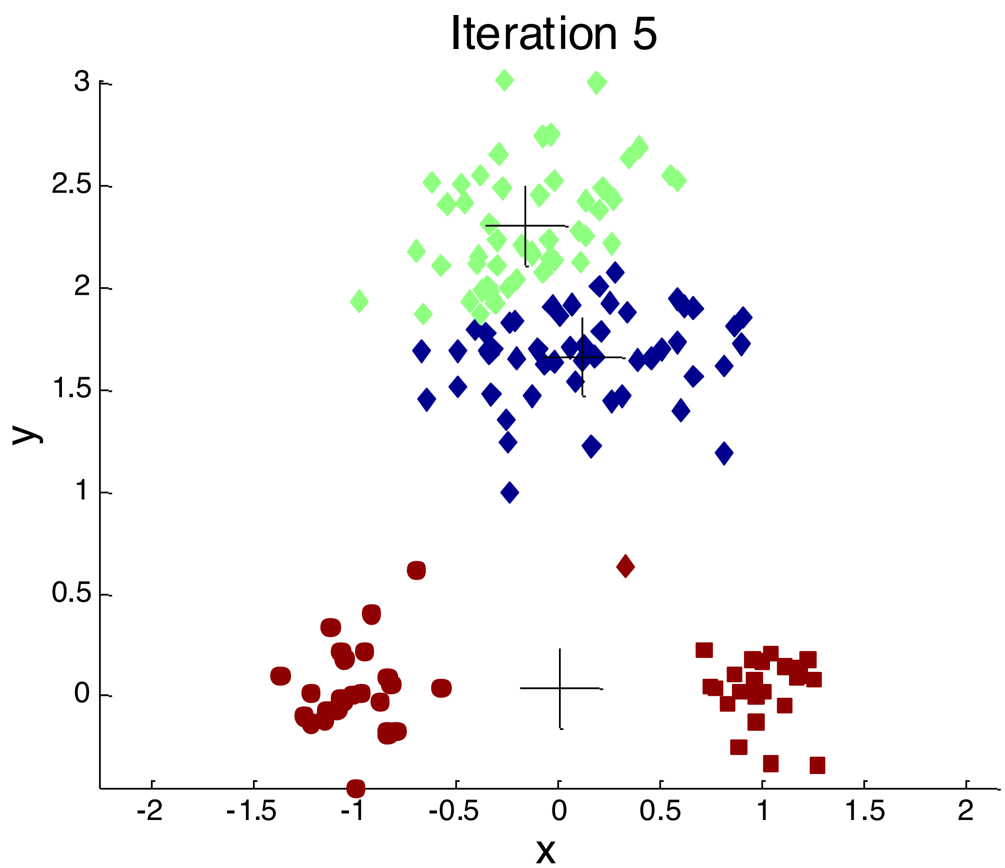


Optimal Clustering

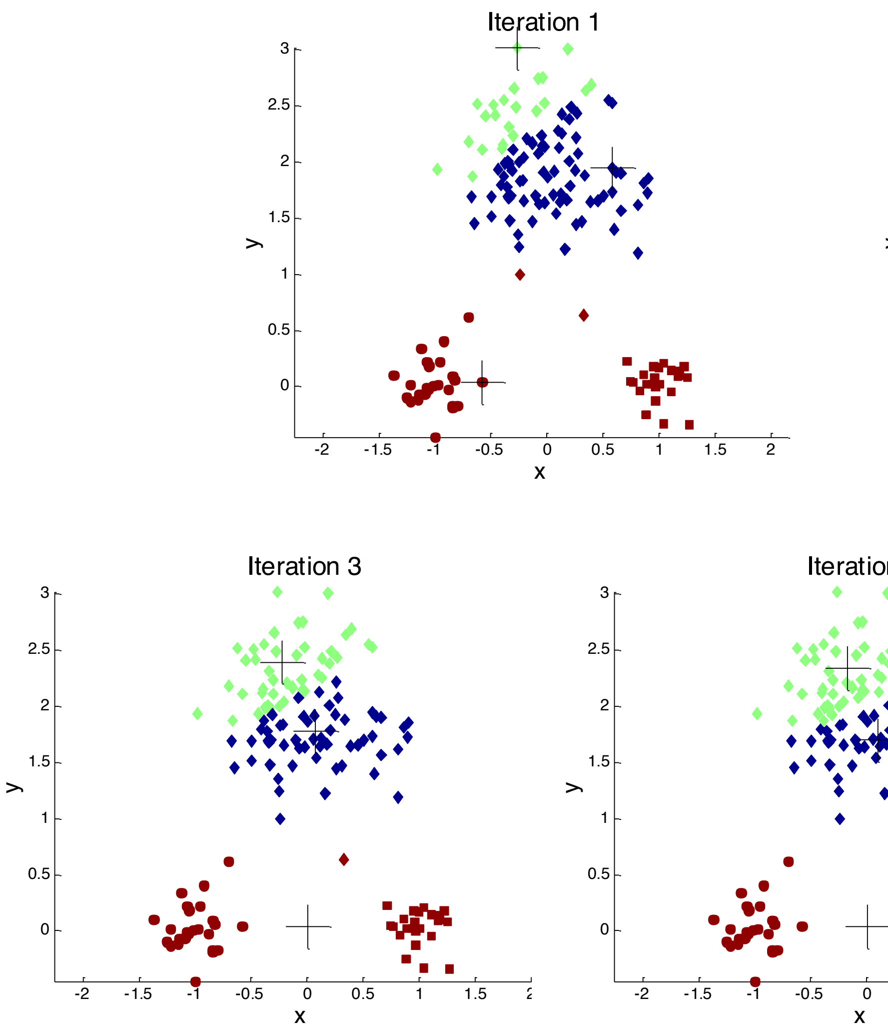
Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …



Trying to find good optimal k-means clusterings

- 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 heirarchical 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](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](http://www.statmethods.net/advstats/cluster.html)

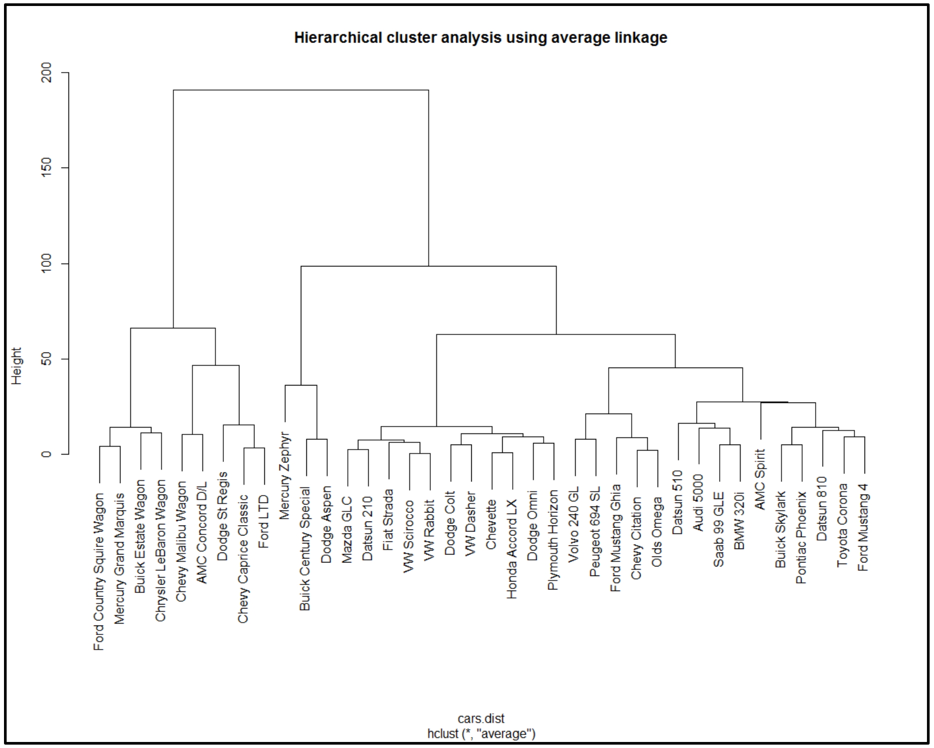
Step 2: Partitioning (if k-means)

```
# 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:
       Buick Estate Wagon Ford Country Squire Wagon        Chevy Malibu Wagon    Chrysler LeBaron Wagon                  Chevette
                        5                         5                         2                         5                         4
           Toyota Corona                Datsun 510                Dodge Omni                Audi 5000             Volvo 240 GL
                        1                         1                         4                         1                         3
             Saab 99 GLE             Peugeot 694 SL      Buick Century Special            Mercury Zephyr              Dodge Aspen
                        1                         3                         2                         3                         2
          AMC Concord D/L      Chevy Caprice Classic                  Ford LTD     Mercury Grand Marquis            Dodge St Regis
                        2                         5                         5                         5                         5
           Ford Mustang 4         Ford Mustang Ghia                 Mazda GLC                Dodge Colt               AMC Spirit
                        1                         3                         4                         4                         1
              VW Scirocco           Honda Accord LX              Buick Skylark            Chevy Citation              Olds Omega
                        4                         4                         1                         3                         3
          Pontiac Phoenix          Plymouth Horizon                Datsun 210               Fiat Strada                 VW Dasher
                        1                         4                         4                         4                         4
               Datsun 810                 BMW 320i                 VW Rabbit
                        1                         1                         4


# get cluster means
aggregate(mydata, by=list(fit$cluster),FUN=mean)

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

[Reference: http://www.statmethods.net/advstats/cluster.html](http://www.statmethods.net/advstats/cluster.html)

Step 3: Visualizing

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



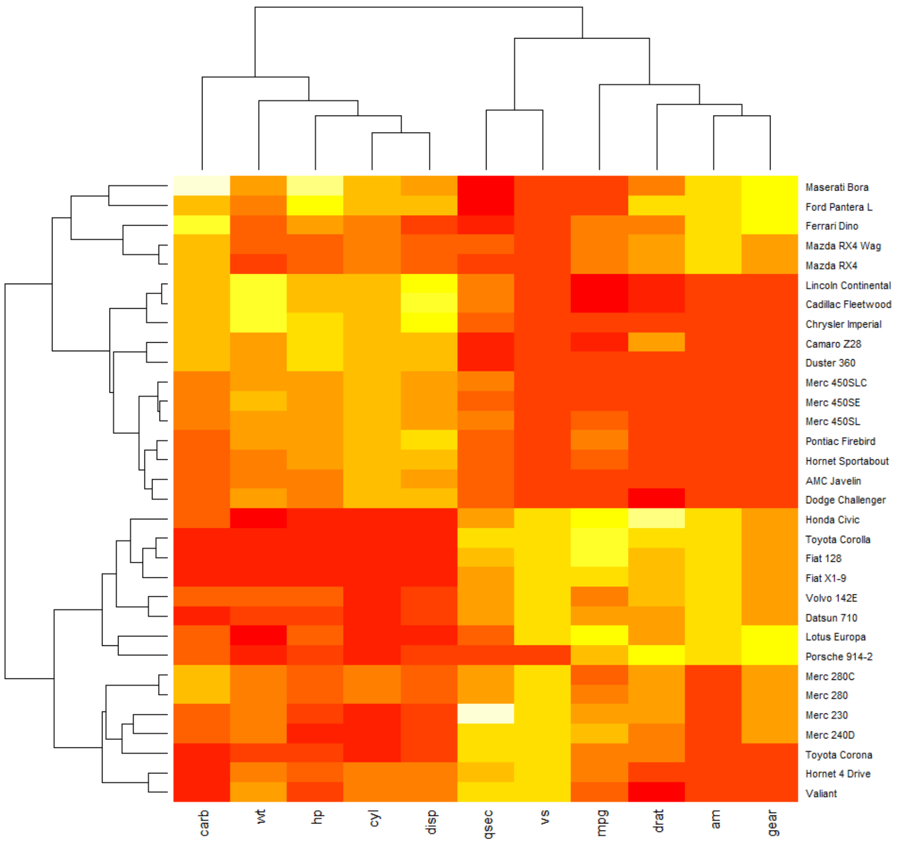
[Reference: http://www.statmethods.net/advstats/cluster.html](http://www.statmethods.net/advstats/cluster.html)

Old Clustering in R: heatmap

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

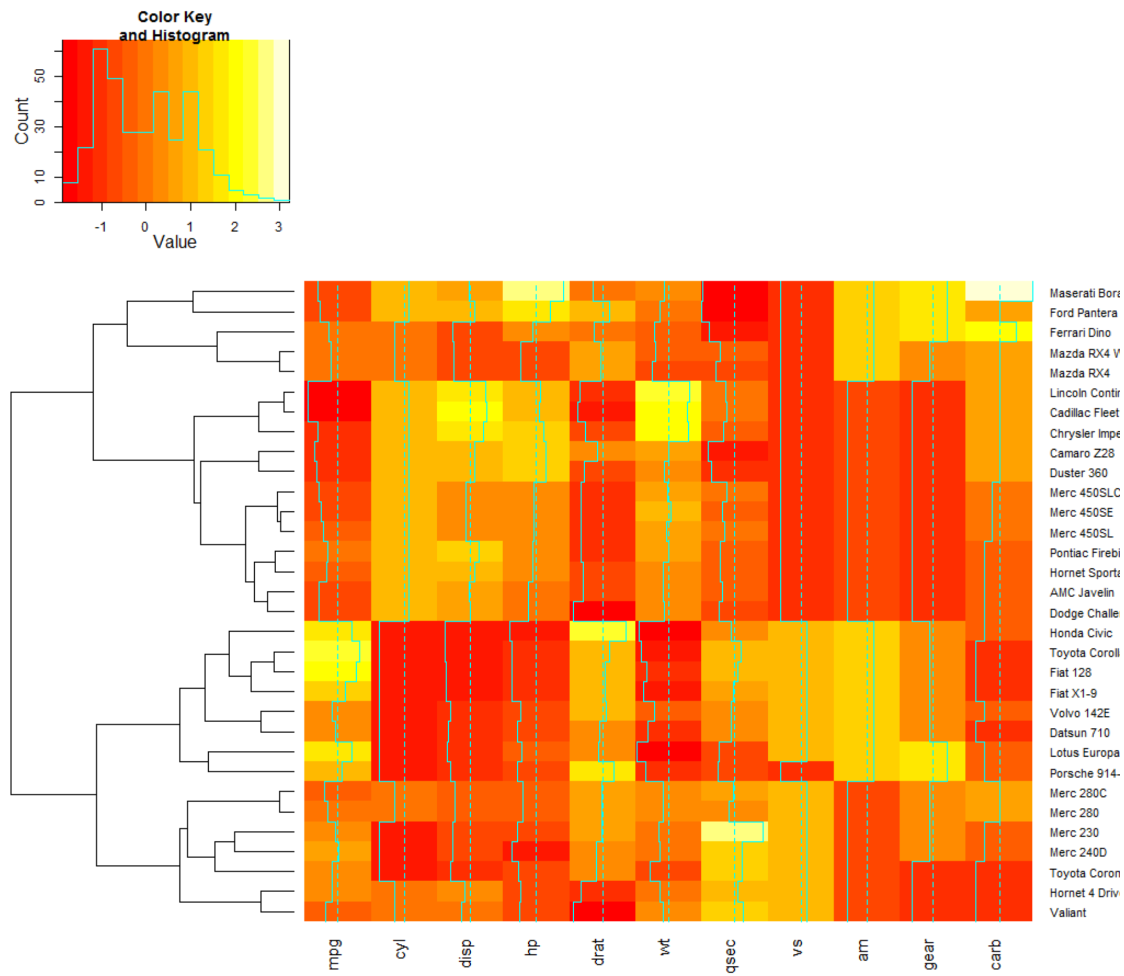
Clustering columns:

- Some info is highly correlated
- e.g. displacement, hp, # cylinders very similar



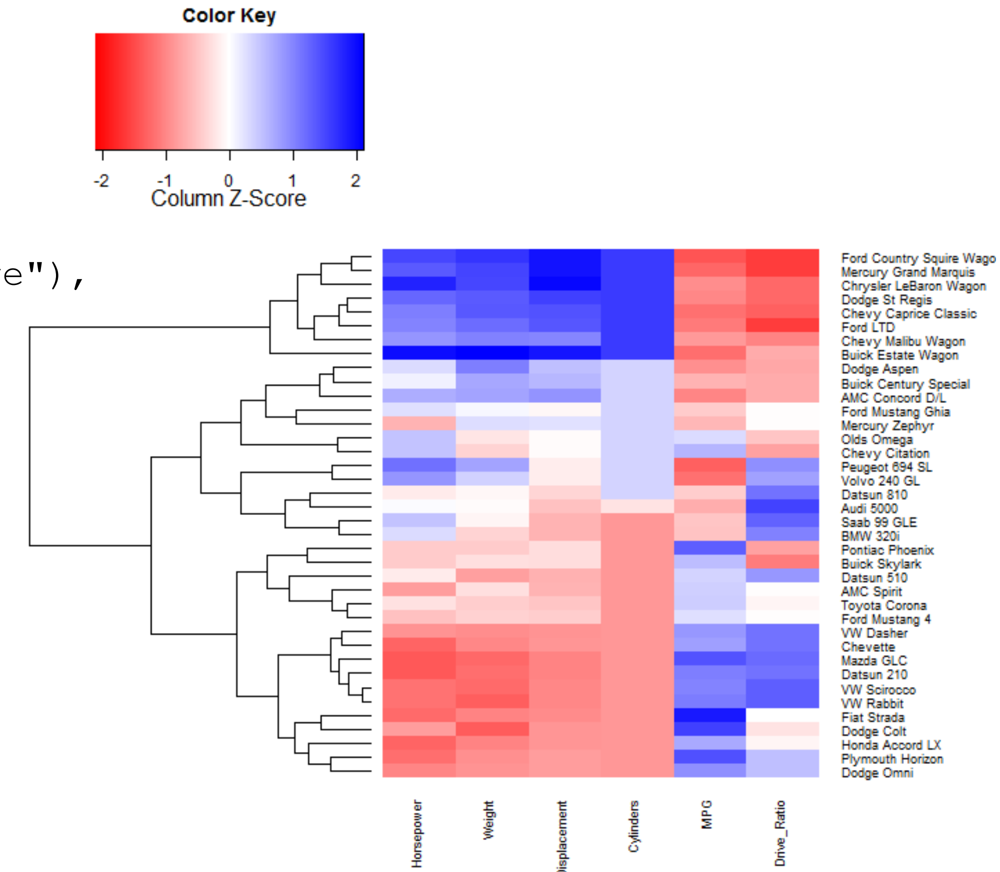
New clustering in R: heatmap.2

```
> 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)
```



Clustering in R: heatmap.2

Reference: http://cran.r-project.org/web/packages/gplots/gplots.pdf

```
heatmap.2 (x,

   # dendrogram control
    Rowv = TRUE,
    Colv=if(symm)"Rowv" else TRUE,
    distfun = dist,
    hclustfun = hclust,
    dendrogram = c("both","row","column","none"),
    symm = FALSE,

    # data scaling
    scale = c("none","row", "column"),
    na.rm=TRUE,

    # image plot
    revC = identical(Colv, "Rowv"),
    add.expr,

    # mapping data to colors
    breaks,
    symbreaks=min(x < 0, na.rm=TRUE) || scale!="none",

    # colors
    col="heat.colors",

    # block sepration
    colsep,
    rowsep,
    sepcolor="white",
    sepwidth=c(0.05,0.05),

    # cell labeling
    cellnote,
    notecex=1.0,
    notecol="cyan",
    na.color=par("bg"),

   # level trace
    trace=c("column","row","both","none"),
    tracecol="cyan",
    hline=median(breaks),
    vline=median(breaks),
    linecol=tracecol,

    # Row/Column Labeling
    margins = c(5, 5),
    ColSideColors,
    RowSideColors,
    cexRow = 0.2 + 1/log10(nr),
    cexCol = 0.2 + 1/log10(nc),
    labRow = NULL,
    labCol = NULL,

    # color key + density info
    key = TRUE,
    keysize = 1.5,
    density.info=c("histogram","density","none"),
    denscol=tracecol,
    symkey = min(x < 0, na.rm=TRUE) || symbreaks,
    densadj = 0.25,

    # plot labels
    main = NULL,
    xlab = NULL,
    ylab = NULL,

    # plot layout
    lmat = NULL,
    lhei = NULL,
    lwid = NULL,
```

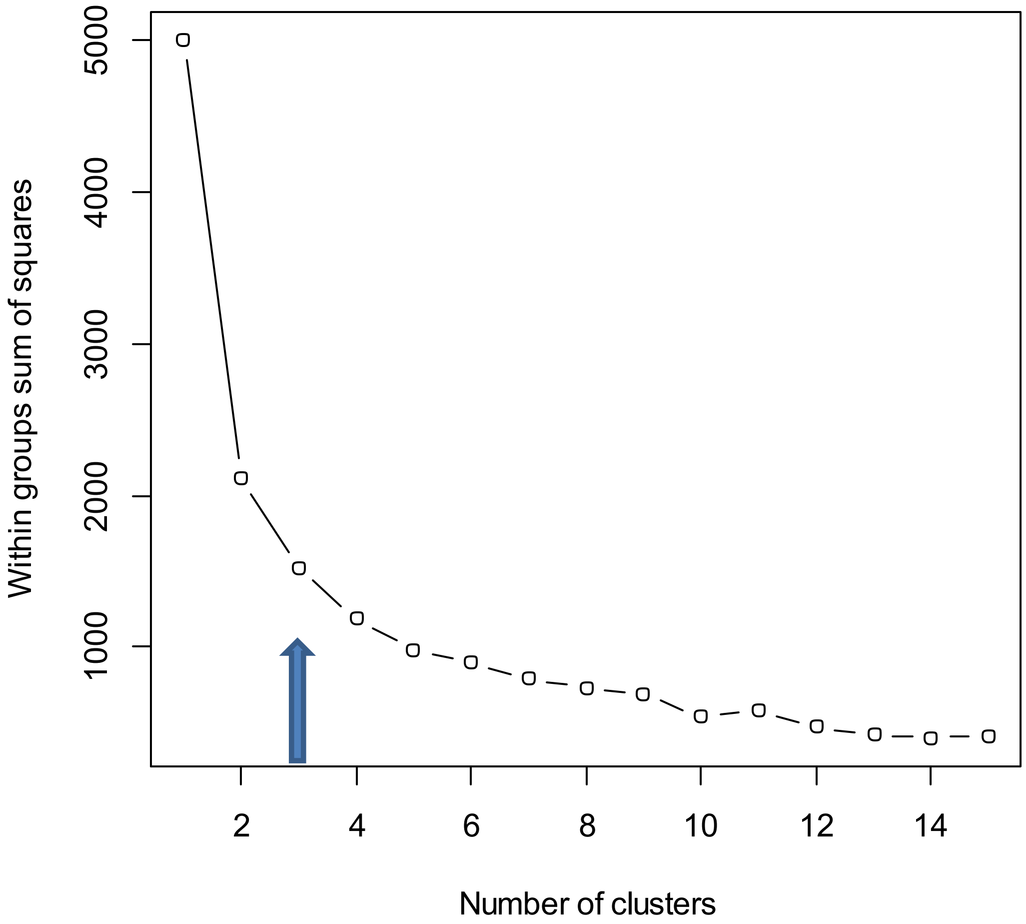
How many clusters?

- 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-heirarchical
  - Ratio of total within-groups variance to between-group variance, vs # of clusters
  - Special case: within-groups sum of squares vs # of clusters
  - Elbow/sharp bend shows point at which adding more clusters helps reduce distortion measure less and less

How many clusters?

```
# Determine number of clusters
wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(mydata,
   centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
  ylab="Within groups sum of squares")
```

How many clusters?



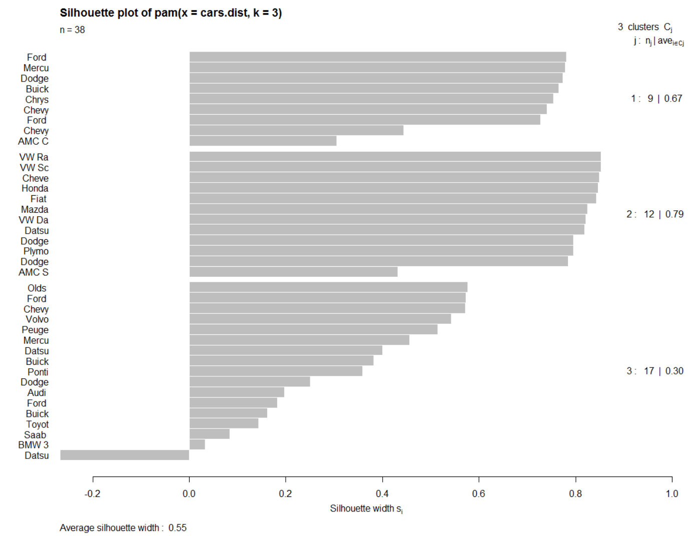
How do we know if we've found good quality clusters?

- 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

Silhouette scores

Peter J. Rousseeuw (1987). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". *Computational and Applied Mathematics* **20**: 53–65.

- A graphical aid for interpretation and validation of cluster analysis
- *a*(*i*) : average dissim of datum *i* with others in same cluster
- *b*(*i*): lowest average dissim for other clusters (neighboring cluster)
- Gives degree of confidence in cluster assignment
  - Well-clustered elements: score near 1
  - Poorly-clustered elements: score near -1 (probably in wrong cluster)
- To compute in R: `silhouette(cars.pam, cars.dist)`



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

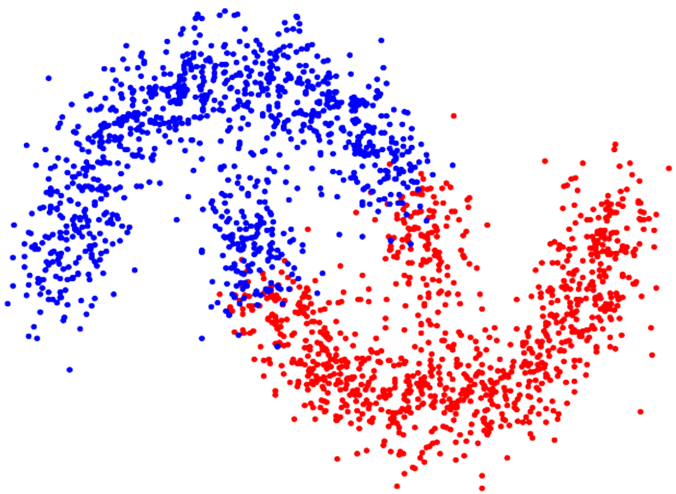
Which can be written as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

See cluster_analysis.R example file

Spectral clustering

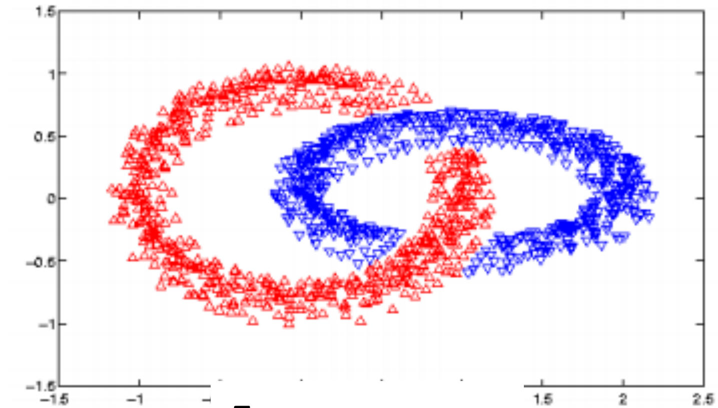
- How does k-means or hierarchical clustering deal with THIS?



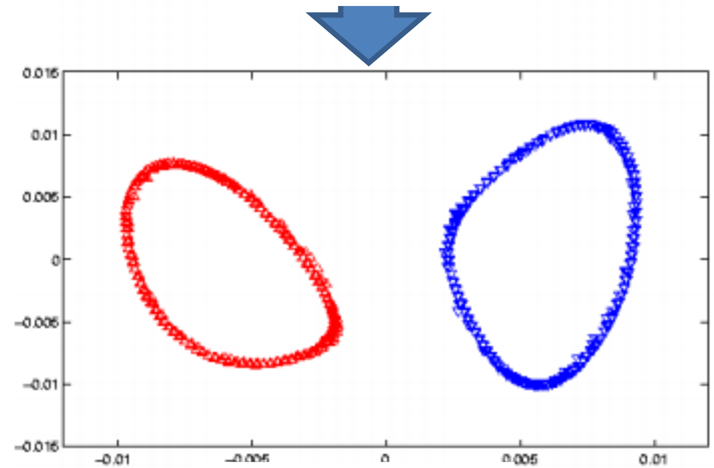
- Not well: data has obvious local structure (lies along curves) but traditional clustering methods don't account for that.

Spectral clustering

1. Map points in regular feature space to ‘spectral’ space
2. Then apply conventional clustering
   – e.g. k-means



Feature space

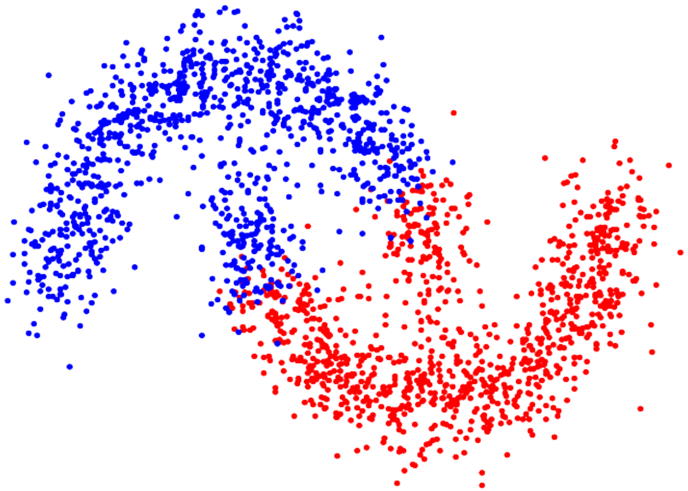


Spectral space

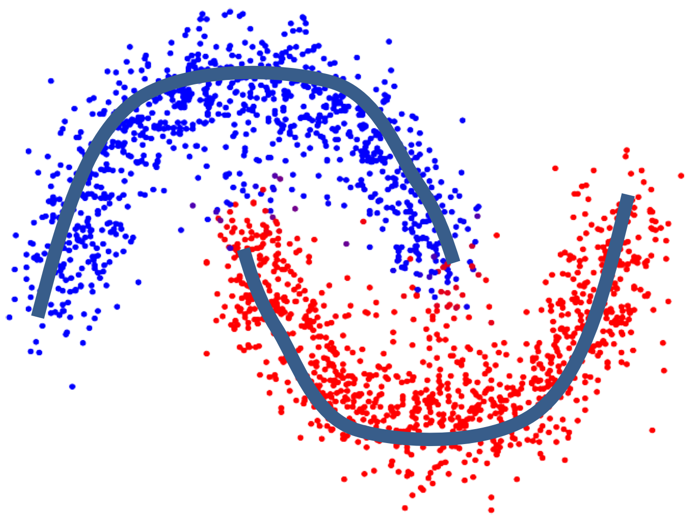
Source: Chris Ding, ICML 2004 Tutorial on Spectral Clustering

Spectral clustering methods work well for data with local geometry structure, i.e. points tend to form curves or surfaces

In case you were curious: The key idea of spectral clustering is to approximate the optimal cluster indicator function by the second eigenvector of the well-known graph Laplacian. It amounts to finding the optimal balanced cut of an undirected weighted graph where the weights represent the similarities between points.



k-means

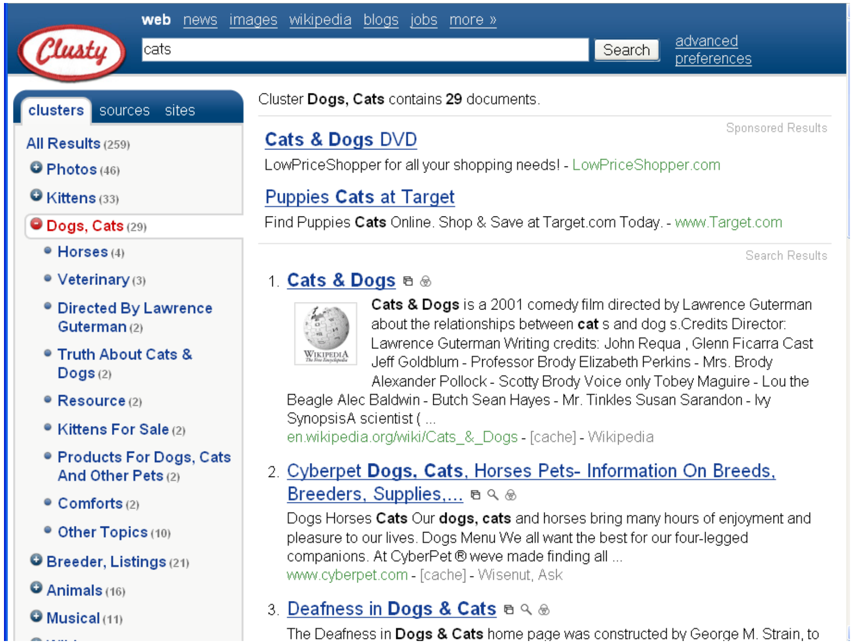


Spectral clustering

Source: http://www.ml.uni-saarland.de/code/pSpectralClustering/pSpectralClustering.html

How to automatically name or label clusters?

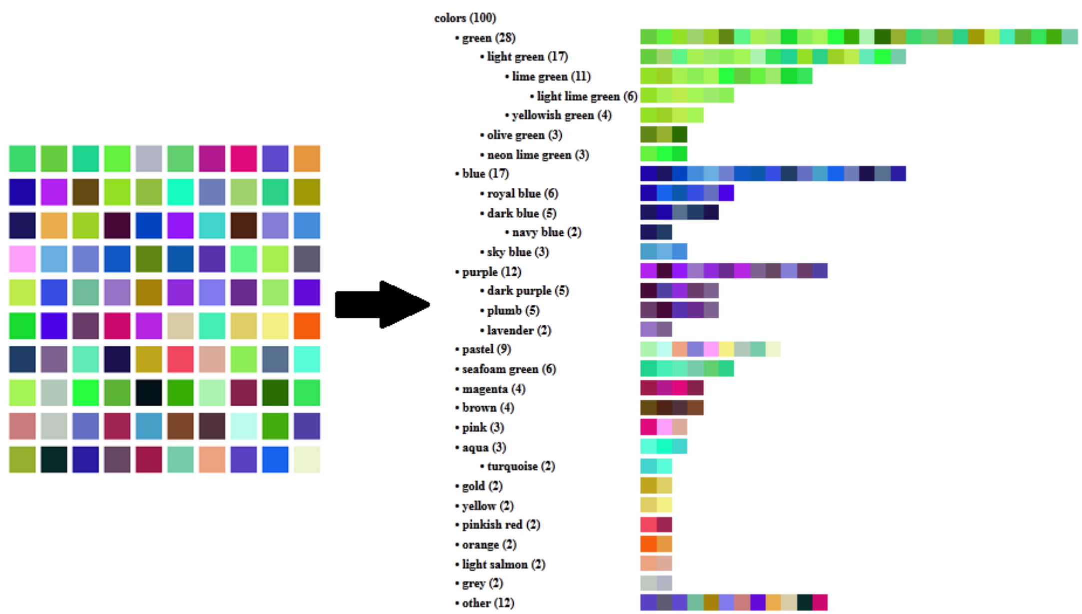
- What determines 'good' vs 'bad' names?
  - Usually task-based evaluations, e.g. search interface effect on user satisfaction/effectiveness
- Use a text summary of a representative element
  - Centroid, medoid, etc.
- Classify with existing heirarchy
- Don't show labels
  - Benefits unclear depending on scenario



See: http://searchuserinterfaces.com/book/sui_ch8_navigation_and_search.html

How to automatically name or label clusters?

- Use crowd-powered methods to create and label object hierarchies [Chilton et al, CHI 2013]
  - And more generally, global pictures of a dataset



[Source: http://hmslydia.com/cascade.html](http://hmslydia.com/cascade.html)

How can we visualize high-dimensional clusters? One method: multi-dimensional scaling (MDS)

- "Flatten" multidimensional cloud to 2d and preserve distances as much as possible

- Input
  - List of N datapoints (*m*-dimensional)
  - Or: derived NxN distance matrix
- Output: list of 2-dimensional datapoints
- Preserve the true relative distances between all pairs of *m*-dimensional points

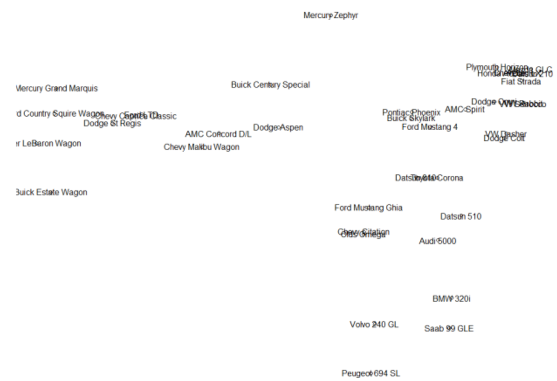
MDS and PAM in R
Partitioning Around Medoids (find k representative objects, then cluster)

- Extract feature points (rows) for objects
- Compute *d* : matrix of distances between rows
- Call cmdscale(d)
- Plot 2-d
-
```
# Classical MDS
# N rows (objects) x p columns (variables)
# each row identified by a unique row name

d <- dist(cars.data) # euclidean distances
between the rows
fit <- cmdscale(d,eig=TRUE, k=2) # k is the
number of dim
fit # view results

# plot solution
x <- fit$points[,1]
y <- fit$points[,2]
plot(x, y, xlab="Coordinate 1",
ylab="Coordinate 2",
  main="Metric MDS", type="n")
text(x, y, labels = rownames(fit$points))
```

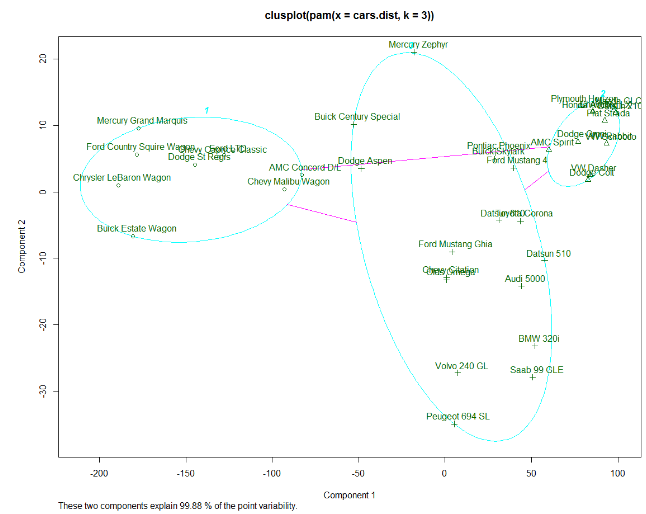
- Also could use
```
  cars.pam = pam(cars.dist, 3)
  clusplot(cars.pam, labels=2)
```

http://stat.ethz.ch/R-manual/R-devel/library/cluster/html/pam.html

MDS



PAM



Summary: Conducting Cluster Analysis

Formulate the Problem

↓

Select a Distance Measure

↓

Select a Clustering Procedure

↓

Decide on the Number of Clusters

↓

Interpret and Profile Clusters

↓

Assess the Validity of Clustering

Homework 4: Cluster Analysis

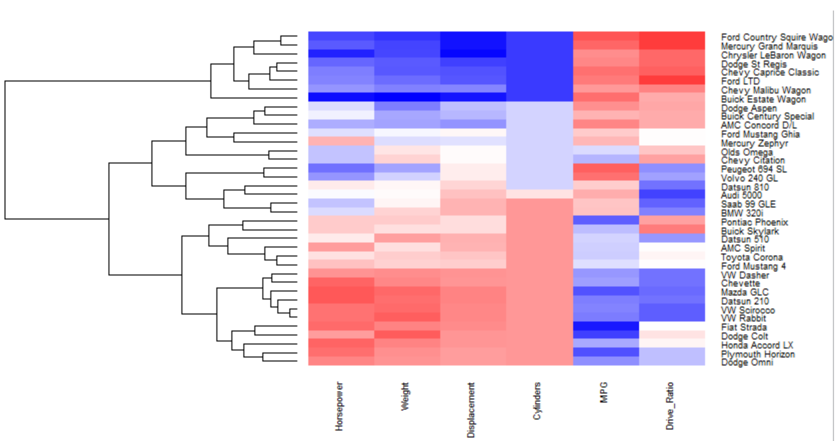
Cars

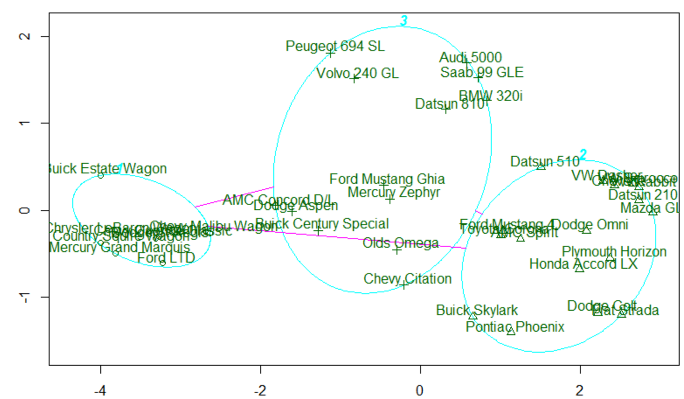
- Cluster analysis of a set of 38 vehicles
- Each row in the data set contains various types of information about each vehicle

What to do:

- Use hierarchical and k-means clustering to cluster car data
- Visualize the results in various ways discussed in the lecture

	Country	Car	MPG	Weight	Drive_Ratio	Horsepower	Displacement	Cylinders
1	U.S.	Buick Estate Wagon	16.9	4.360	2.73	155	350	8
2	U.S.	Ford Country Squire Wagon	15.5	4.054	2.26	142	351	8
3	U.S.	Chevy Malibu Wagon	19.2	3.605	2.56	125	267	8
4	U.S.	Chrysler LeBaron Wagon	18.5	3.940	2.45	150	360	8
5	U.S.	Chevette	30.0	2.155	3.70	68	98	4
6	Japan	Toyota Corona	27.5	2.560	3.05	95	134	4
7	Japan	Datsun 510	27.2	2.300	3.54	97	119	4
8	U.S.	Dodge Omni	30.9	2.230	3.37	75	105	4
9	Germany	Audi 5000	20.3	2.830	3.90	103	131	5
10	Sweden	Volvo 240 GL	17.0	3.140	3.50	125	163	6





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