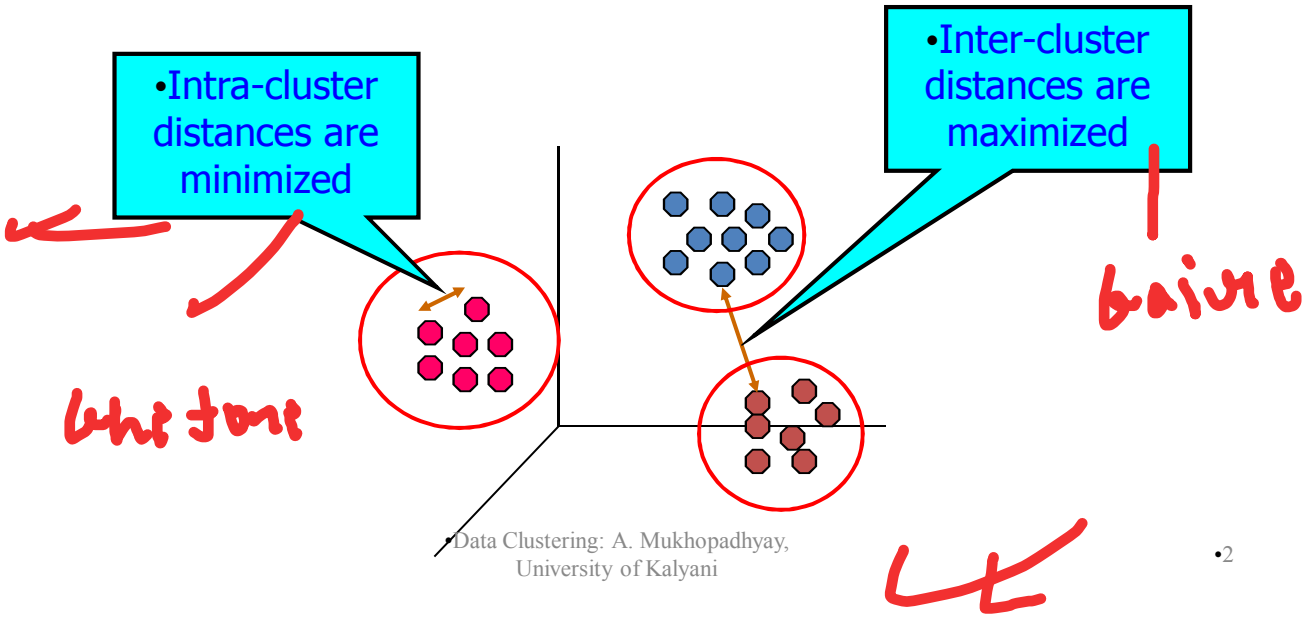
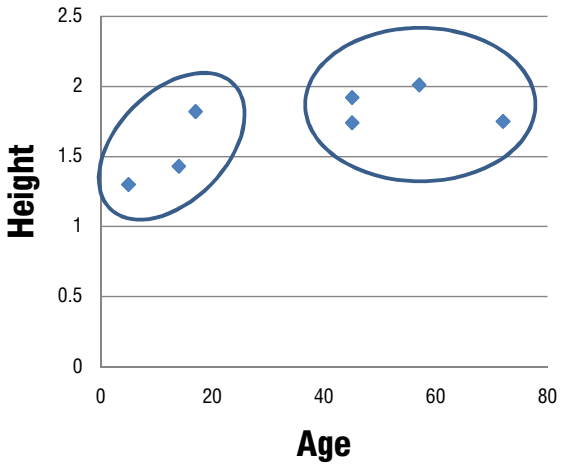
What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Dataset for Clustering

Person	Age (Y)	Height (m)
A	32	1.82
B	45	1.74
C	14	1.43
D	02	0.82
E	82	1.75
F	57	2.01
G	45	1.92



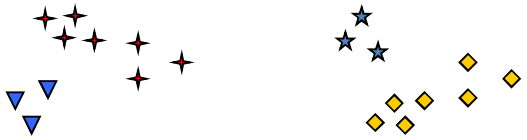
- **The dataset may be high dimensional with many attributes like age, height, weight, sex, hair color, … …**
  - **Scatter plot like above not possible.**
  - **No idea of how many clusters**

Notion of a Cluster can be Ambiguous

•How many clusters?

•Six Clusters

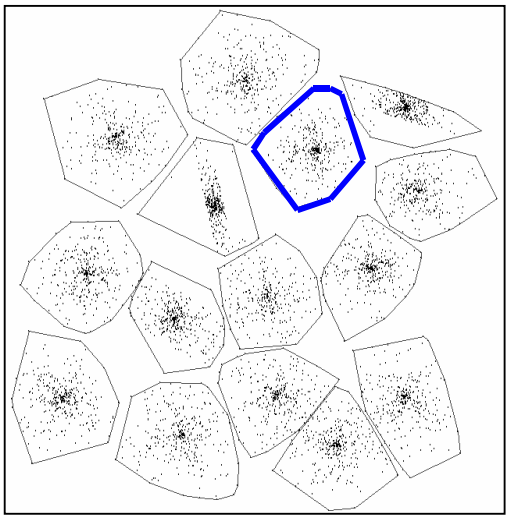
•Two Clusters



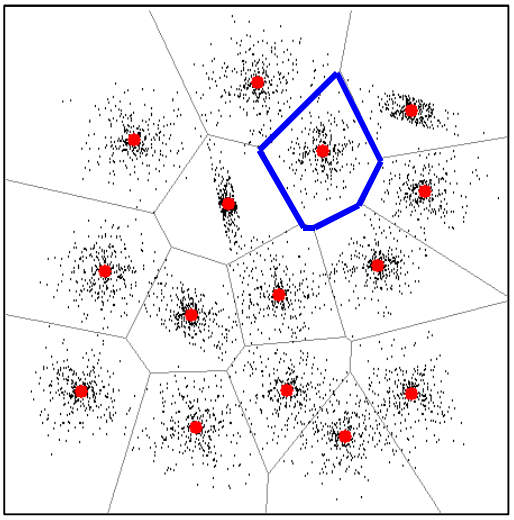
•Four Clusters

Clustering result as partition

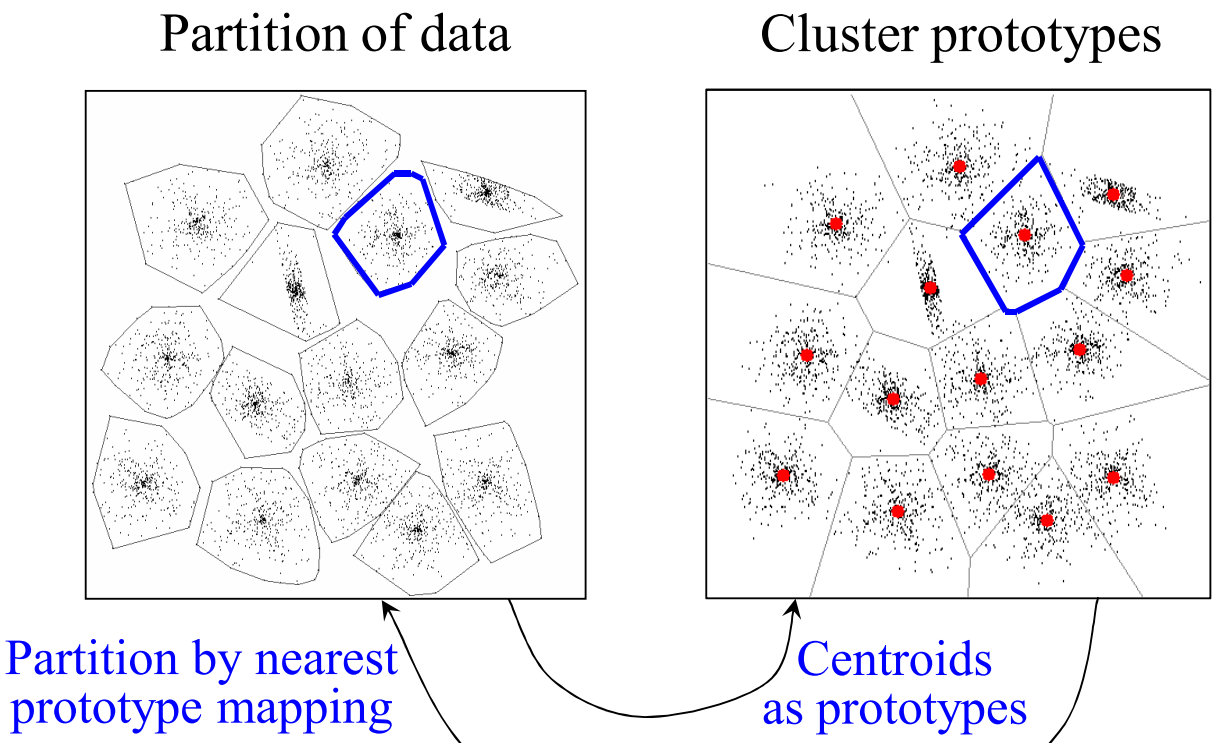
Partition of data



Cluster prototypes



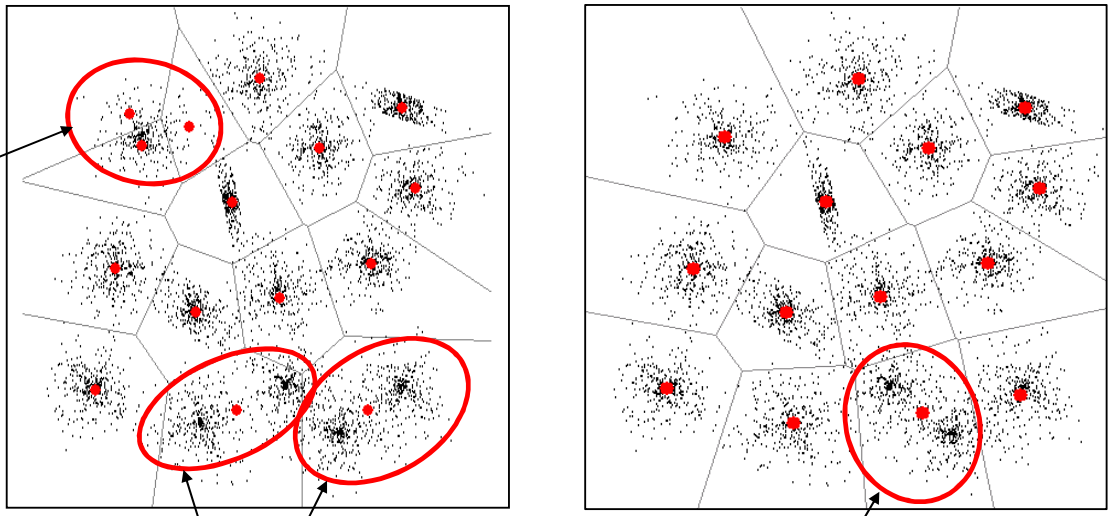
Duality of partition and centroids



Challenges in clustering

Incorrect cluster allocation

Incorrect number of clusters



How to solve?

Solve the clustering:

- Given input data (X) of N data vectors, and number of clusters (K), find the clusters.
- Result given as a set of prototypes, or partition.

Algorithmic problem

Solve the number of clusters:

- Define appropriate cluster validity function f.
- Repeat the clustering algorithm for several M.
- Select the best result according to f.

Mathematical problem

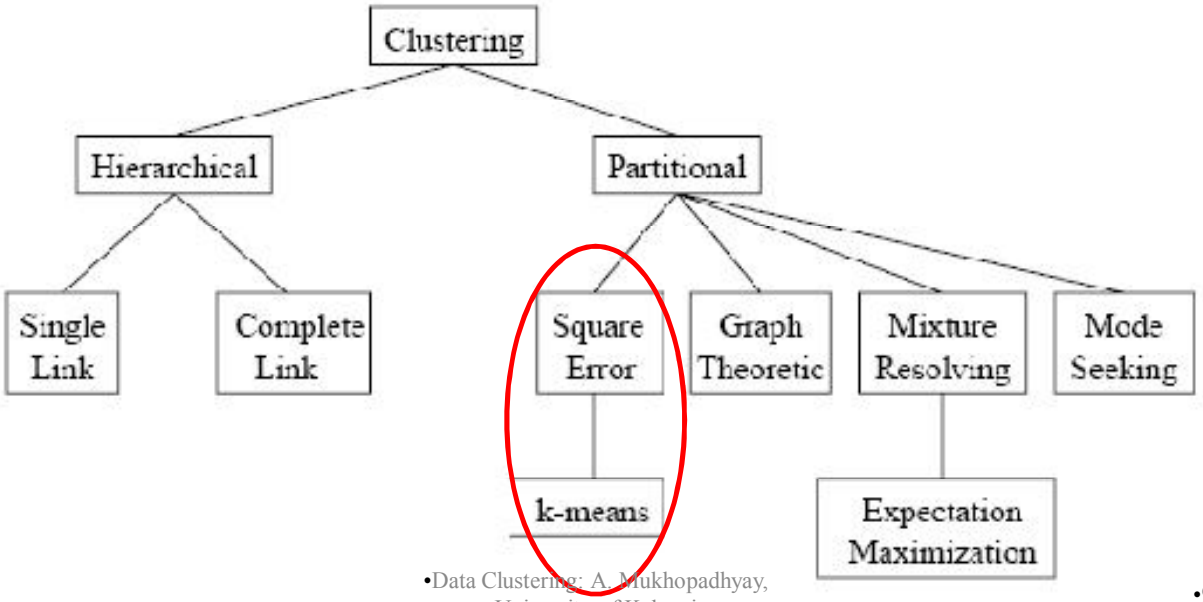
Solve the problem efficiently.

Computer science problem

Taxonomy of clustering

[Jain, Murty, Flynn, Data clustering: A review, *ACM Computing Surveys*, 1999.]

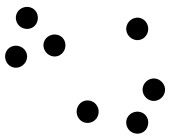
- One possible classification based on cost function.
- MSE is well defined and most popular.



Definitions and data

Set of N data points:

$$X=\{x_1, x_2, \ldots, x_N\}$$

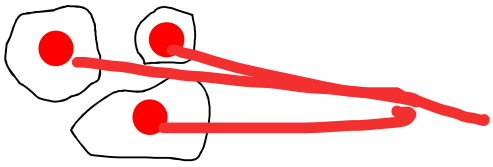


Partition of the data:

$$P=\{p_1, p_2, \ldots, p_M\},$$

Set of M cluster prototypes (centroids):

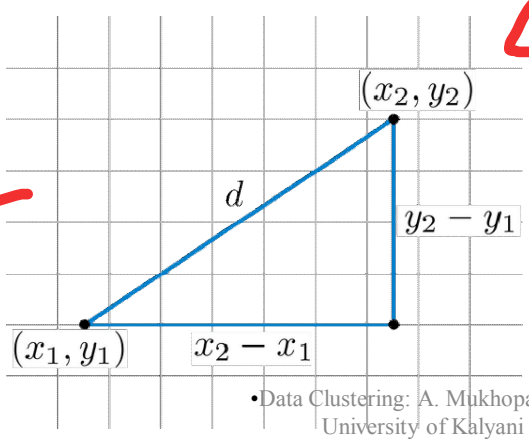
$$C=\{c_1, c_2, \ldots, c_M\},$$



Distance function

Euclidean distance of data vectors:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} \left(x_i^k - x_j^k\right)^2}$$



Dependency of data structures

- **Centroid condition**: for a given partition (P), optimal cluster centroids (C) for minimizing MSE are the average vectors of the clusters:

$$c_j = \frac{\sum_{p_i=j} x_i}{\sum_{p_i=j} 1} \quad \forall\ j \in [1, K]$$

- **Optimal partition**: for a given centroids (C), optimal partition is the one with nearest centroid :

$$p_i = \arg\min_{1 \le j \le M} d(x_i, c_j)^2 \quad \forall\ \ i \in [1, N]$$

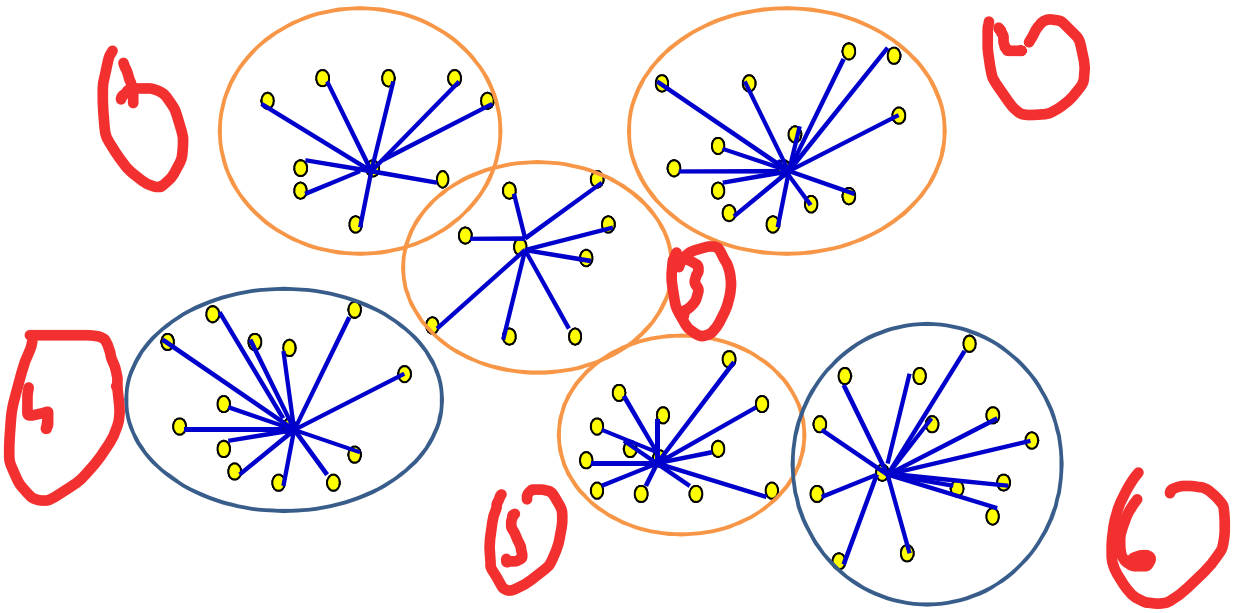
Partitional Clustering & Cost Function

- Partitioning Clustering Approach
  - a typical clustering analysis approach via iteratively partitioning training data set to learn a partition of the given data space
  - learning a partition on a data set to produce several non-empty clusters (usually, the number of clusters given in advance)
  - in principle, optimal partition achieved via minimising the sum of squared distance to its "representative object" in each cluster

$$J = \Sigma_{k=1}^{K} \Sigma_{\mathbf{x} \in C_k} d^2(\mathbf{x}, \mathbf{m}_k)$$

Cost Function

$$J = \Sigma_{k=1}^{K} \Sigma_{\mathbf{x} \in C_k} d^2(\mathbf{x}, \mathbf{m}_k)$$



K-means Clustering - Introduction

- Given a *K*, find a partition of *K clusters* to optimise the chosen partitioning criterion (cost function)
  - global optimum: exhaustively search all partitions
- The *K-means* algorithm: a heuristic method
  - K-means algorithm (MacQueen'67): each cluster is represented by the centre of the cluster and the algorithm converges to stable centriods of clusters.
  - K-means algorithm is the simplest partitioning method for clustering analysis and widely used in data mining applications.

K-means Algorithm

- Given the cluster number *K*, the *K-means* algorithm is carried out as follows:

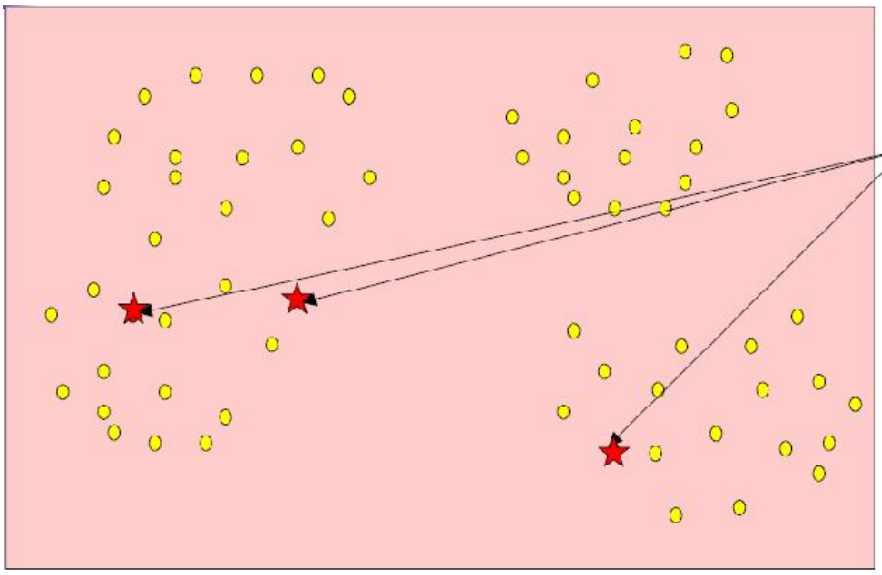
1) Initialisation: Choose K random points as initial cluster centroids.

2) Assign each object to the cluster of the nearest clutser center measured with a specific distance metric

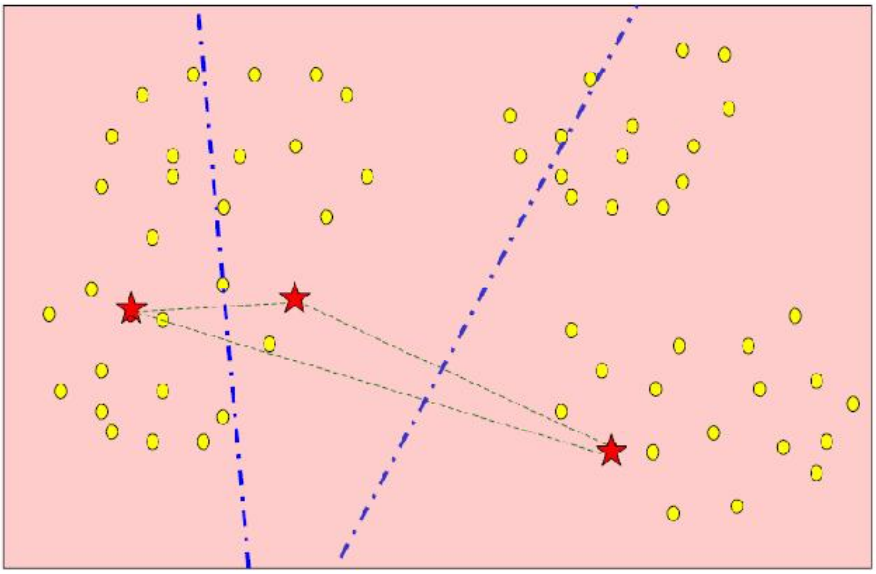
3) Update each cluster center with the **means** of the points assigned to that cluster.

4) Go back to Step 2), stop when no more new assignment (i.e., membership in each cluster no longer changes)

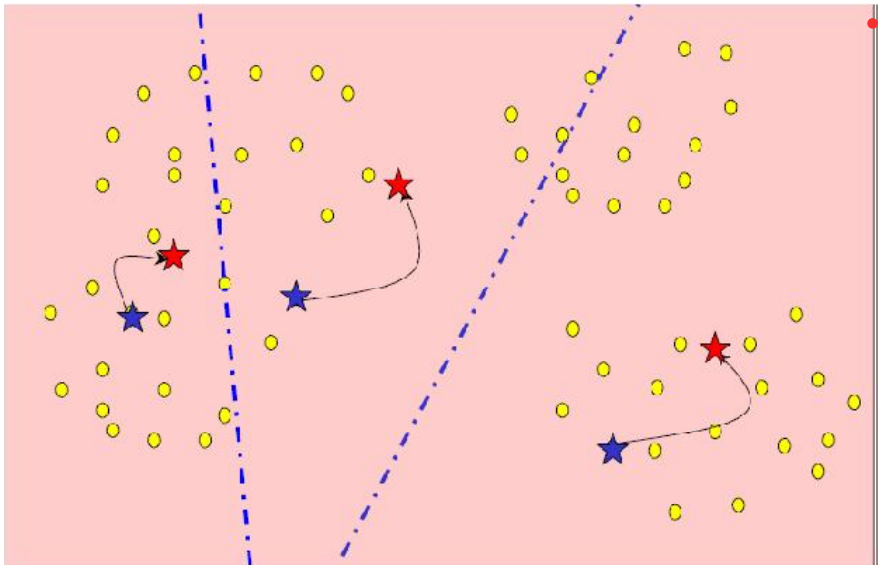
K-means - Example Initialization



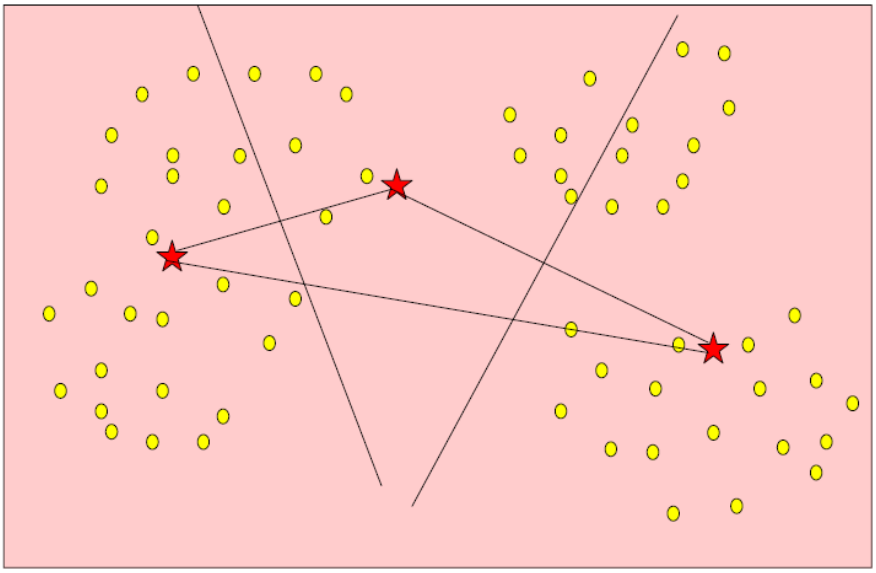
K-means - Example (Contd.)
Assign points



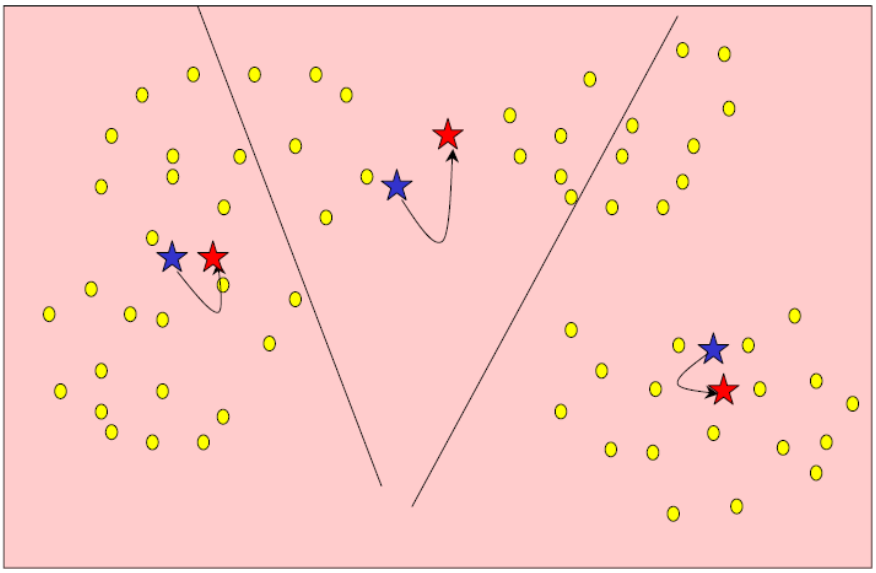
K-means - Example (Contd.)
Update centers



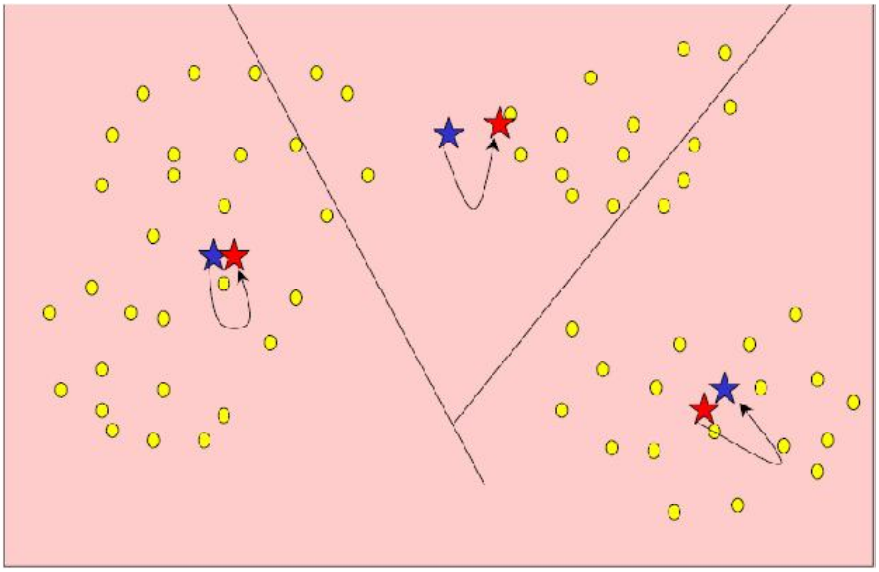
K-means - Example (Contd.)
Reassign points



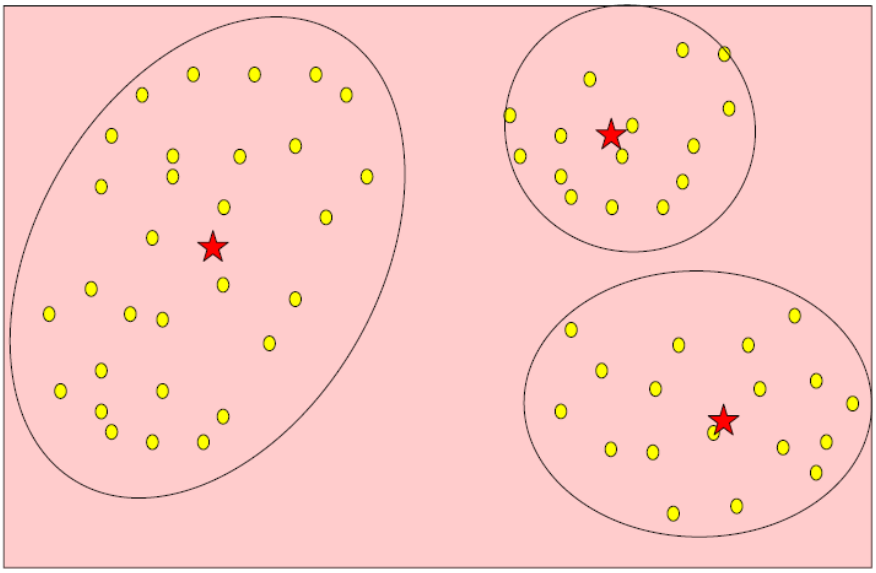
K-means - Example (Contd.)
Update centers



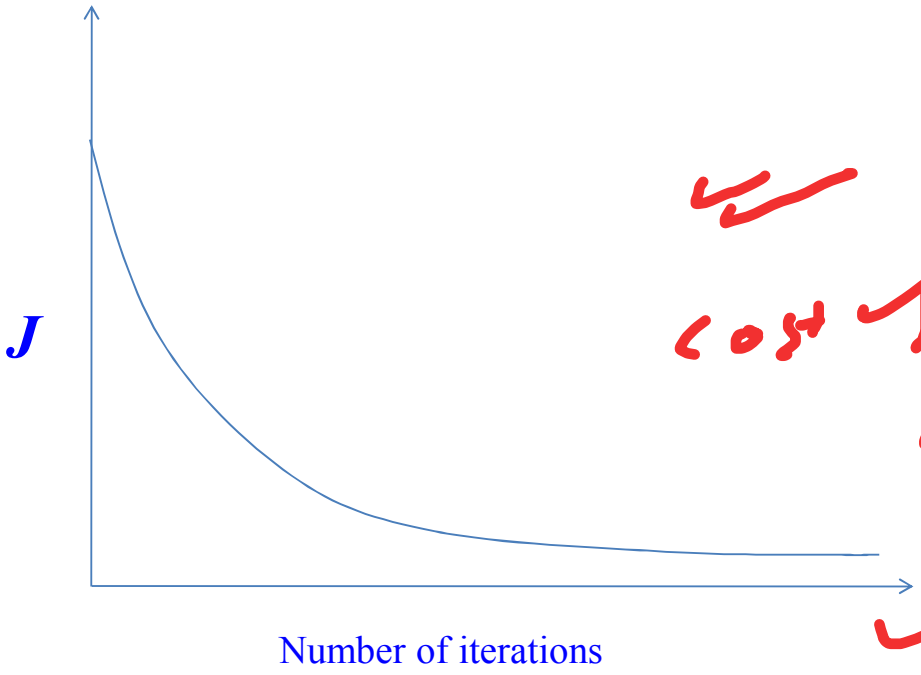
K-means - Example (Contd.)
Reassign points and update centers



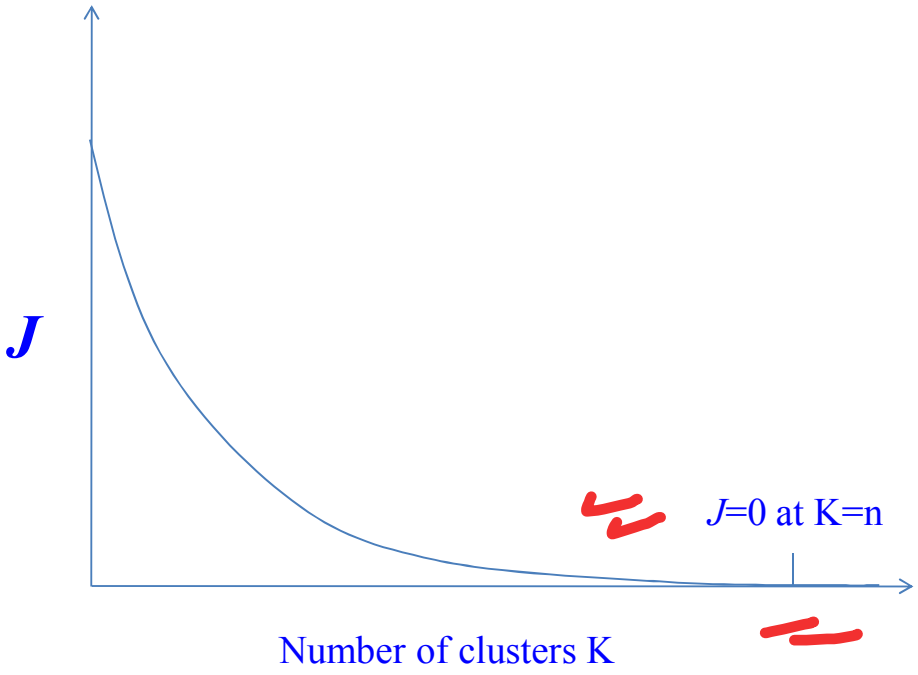
K-means - Example (Contd.)
Final clustering after few iterations



Number of iterations vs. *J*



Number of clusters vs. *J*



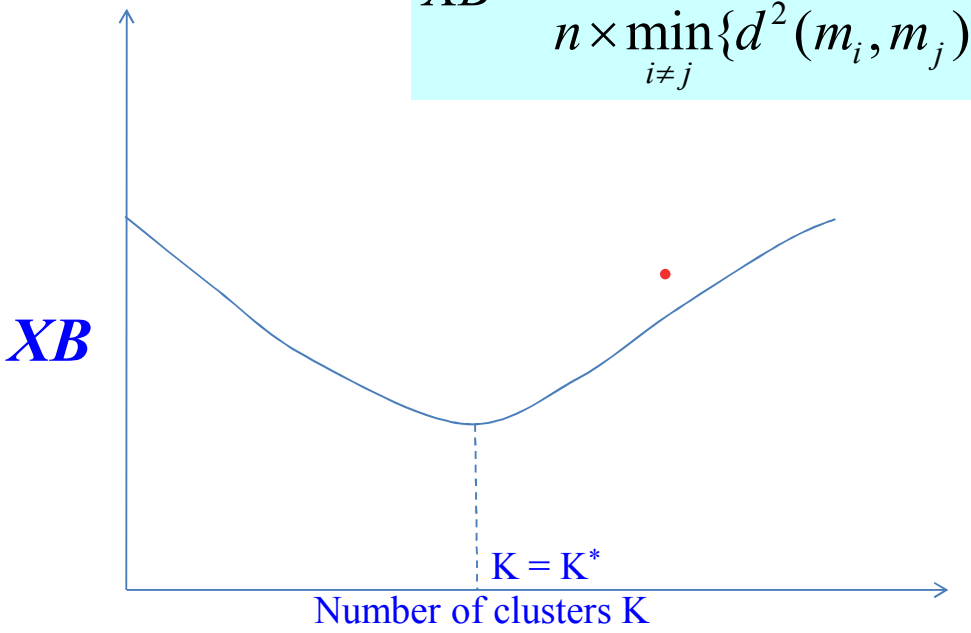
K-means Disadvantages

- K must be known.
  - Not possible for many real-life data.
  - Solution: validity index
- Affected by outliers.
  - Solution: K-medoids clustering
- Gives suboptimal value of J depending on the initial choice of centers.
  - Solution: Repeated runs
- Cannot work on **non-numeric** data.
  - Solution: K-medoids, K-modes
- Work for round shaped and similar sized clusters only.
  - Solution: hierarchical and density-based clustering.

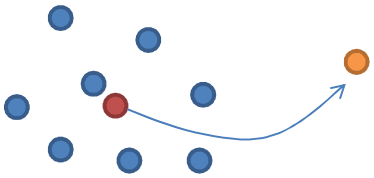
Detecting Number of Clusters using Validity Index

- Xie-Beni Index

$$XB = \frac{\Sigma_{k=1}^{K} \Sigma_{\mathbf{x} \in C_k} d^2(\mathbf{x}, \mathbf{m}_k)}{n \times \min_{i \neq j} \{d^2(m_i, m_j)\}}$$

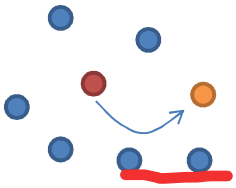


Effect of Outlier



Shifting of cluster mean

Medoid – Most centrally located point of a cluster



Shifting of cluster medoid

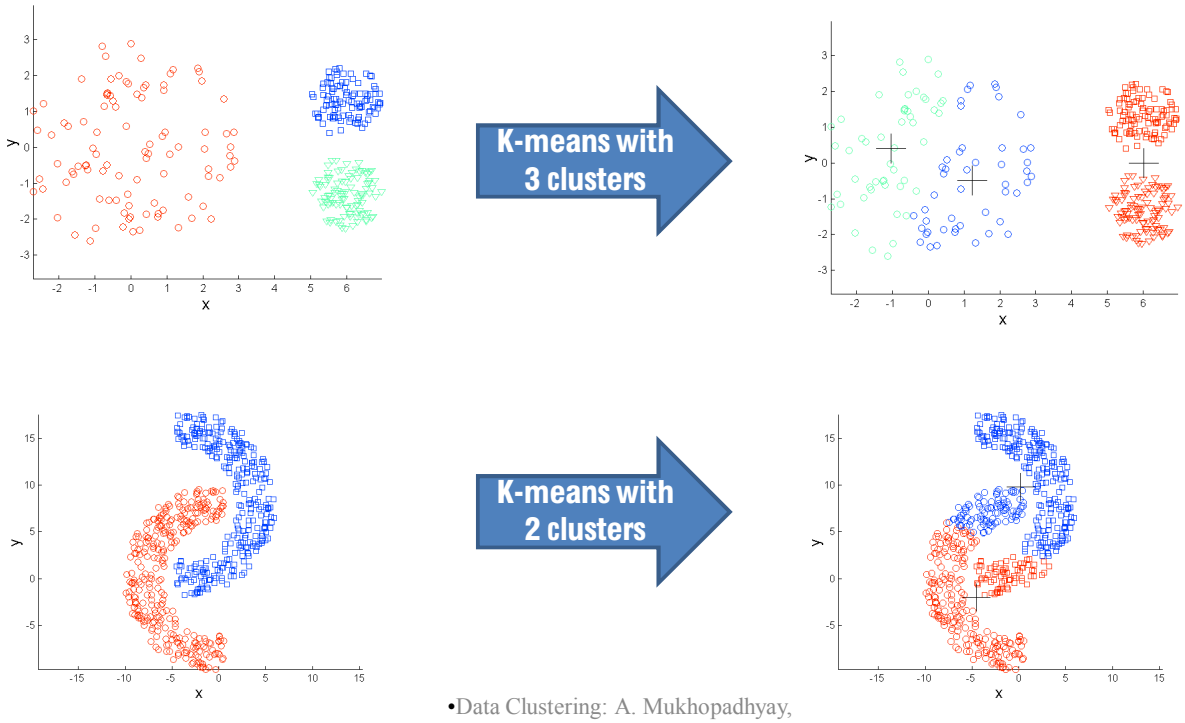
Suboptimal value of J

- For each K, run K-means with different initial sets of cluster centers.
- Find the clustering with minimum J for a particular K.
- Repeat this for different K values.
- Choose the final K and clustering based on some cluster validity index (e.g., XB index).

Non-numeric Data

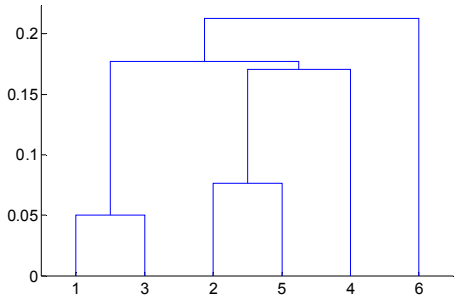
- K-means can work only for features with numeric values, because we need to **compute the means**.
- If the feature is **categorical**, e.g. – color of hair, religion, etc., **mean computation is not possible.**
- For categorical data, K-medoids algorithm can be used with suitable distance function (**Euclidean distance is not computable**).

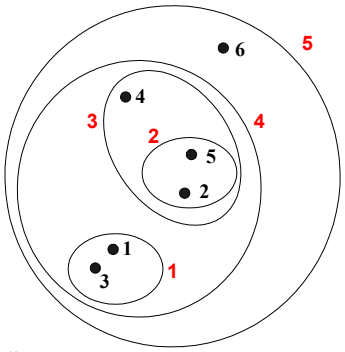
K-means for clusters with different sizes and shapes



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

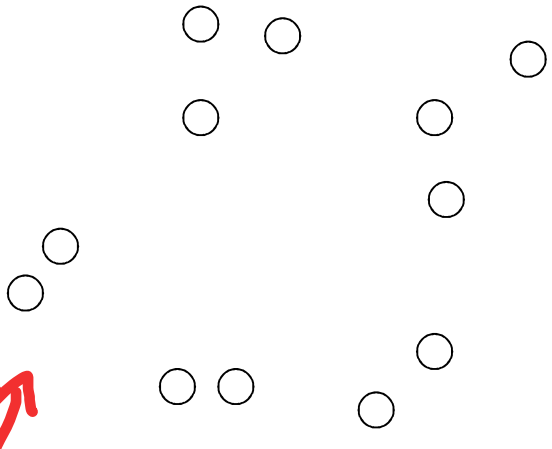
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative Clustering Algorithm

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

Starting Situation

- Start with clusters of individual points and a Distance matrix



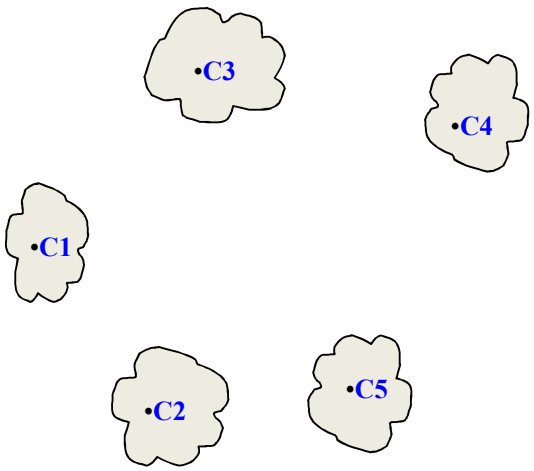
	•p1	•p2	•p3	•p4	•p5	. .
•p1						.
•p2						
•p3						
•p4						
•p5						
.						
.						
.						

•Distance Matrix

Individual clusters

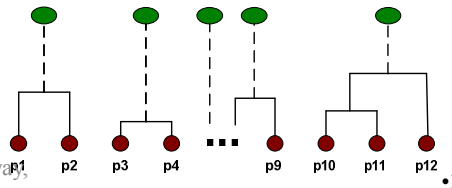
Intermediate Situation

- After some merging steps, we have some clusters



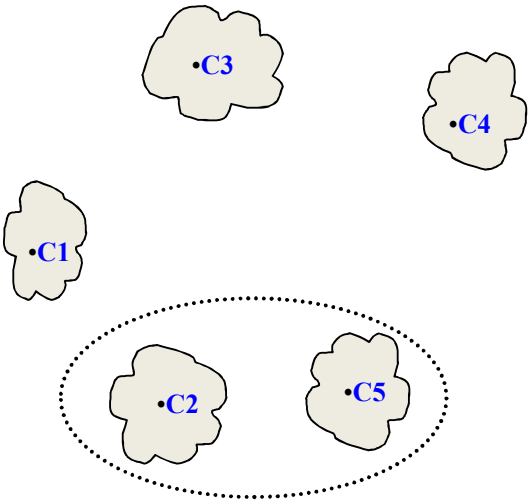
	•C1	•C2	•C3	•C4	•C5
•C1					
•C2					
•C3					
•C4					
•C5					

•Distance Matrix



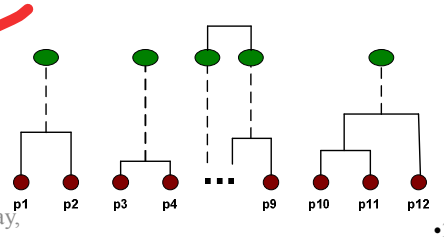
Intermediate Situation

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



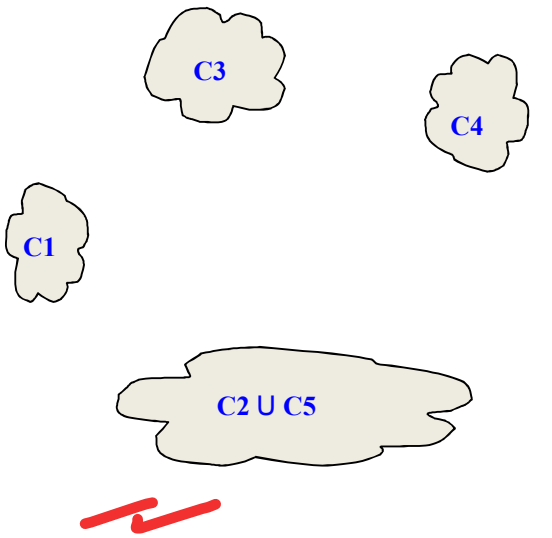
	•C1	•C2	•C3	•C4	•C5
•C1					
•C2					
•C3					
•C4					
•C5					

•Distance Matrix



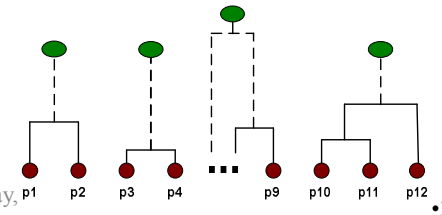
After Merging

The question is "How do we update the Distance matrix?"

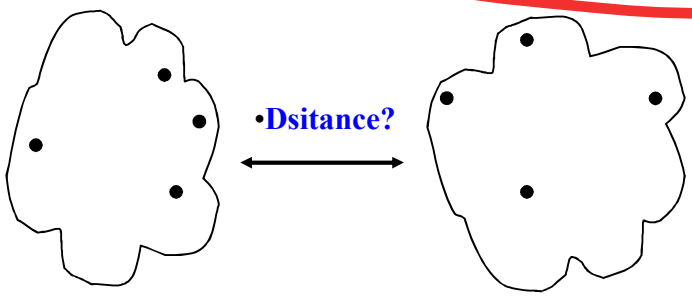


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Distance Matrix



How to Define Inter-Cluster Similarity

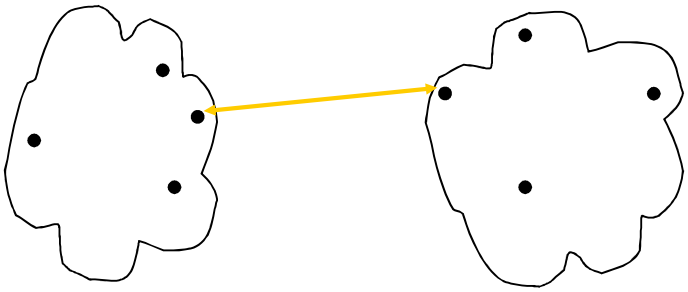


	•p1	•p2	•p3	•p4	•p5	•. . .
•p1						
•p2						
•p3						
•p4						
•p5						
•.						
•.						
•.						

Distance Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

How to Define Inter-Cluster Similarity

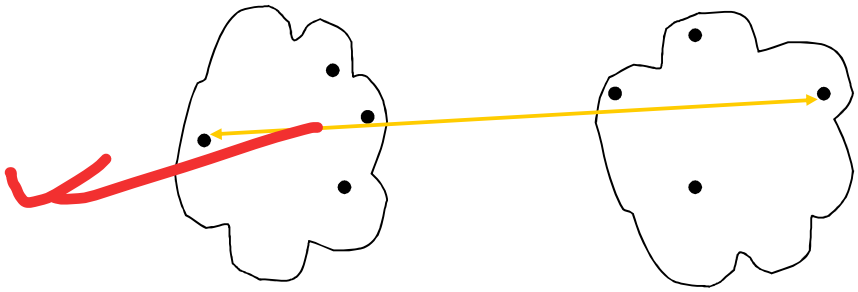


	•p1	•p2	•p3	•p4	•p5	•. . .
•p1						
•p2						
•p3						
•p4						
•p5						
•.						
•.						
•.						

- MIN (Single linkage)
- MAX
- Group Average
- Distance Between Centroids

•**Distance Matrix**

How to Define Inter-Cluster Similarity

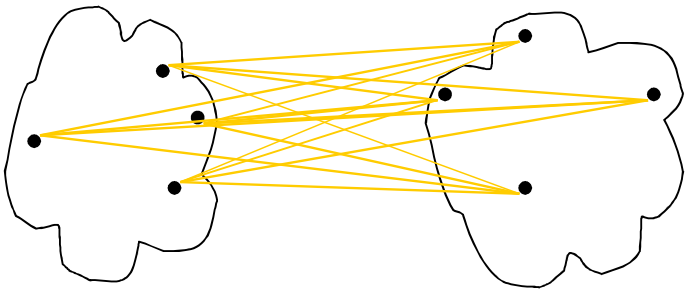


	•p1	•p2	•p3	•p4	•p5	•. . .
•p1						
•p2						
•p3						
•p4						
•p5						
•.						
•.						
•.						

•**Distance Matrix**

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

How to Define Inter-Cluster Similarity

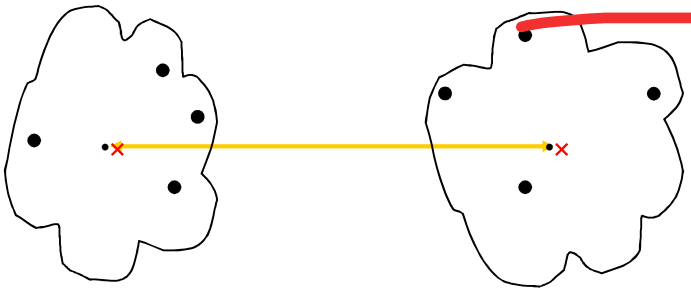


	•p1	•p2	•p3	•p4	•p5	•. . .
•p1						
•p2						
•p3						
•p4						
•p5						
•.						

- MIN
- MAX
- Group Average (Average Linkage)
- Distance Between Centroids

•Distance Matrix

How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids
  - (Centroid Linkage)

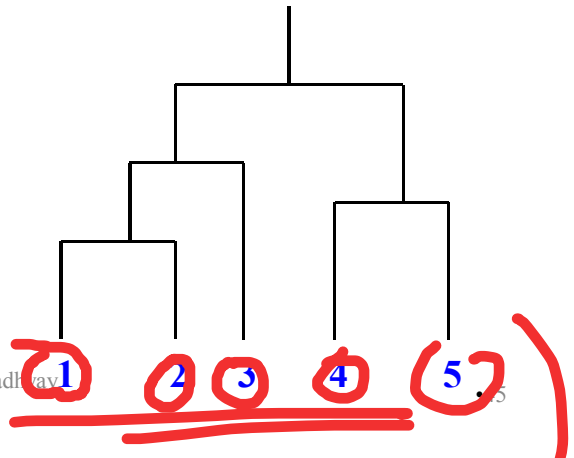
	•p1	•p2	•p3	•p4	•p5	•. . .
•p1						
•p2						
•p3						
•p4						
•p5						
•.						
•.						
•.						

•**Distance Matrix**

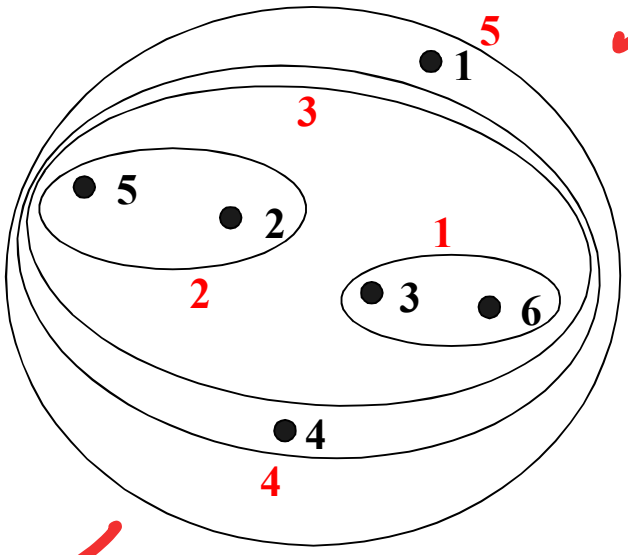
Cluster Distance: MIN or Single Linkage

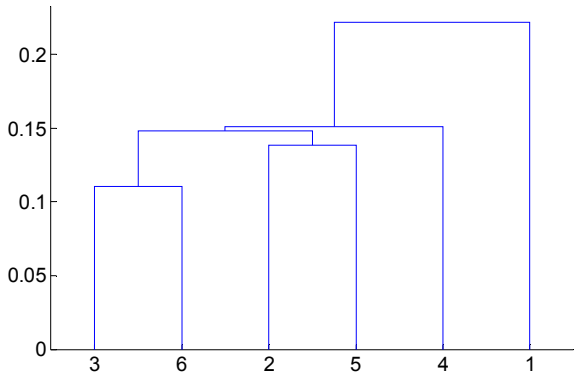
- Distance of two clusters is based on the two most similar (closest) or least-distant points in the different clusters
  - Determined by one pair of points, i.e., by one link in the Distance graph.

	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

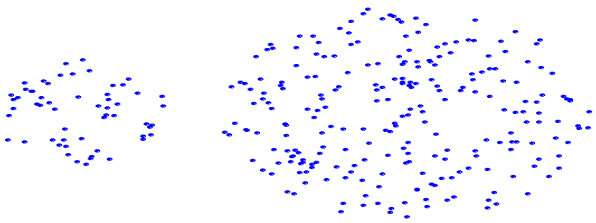


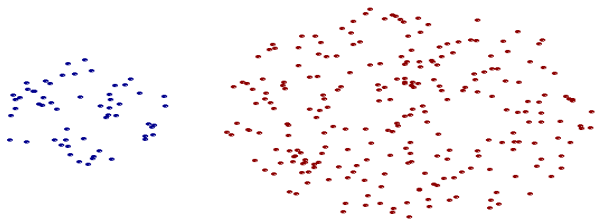


•**Nested Clusters**

•**Dendrogram**

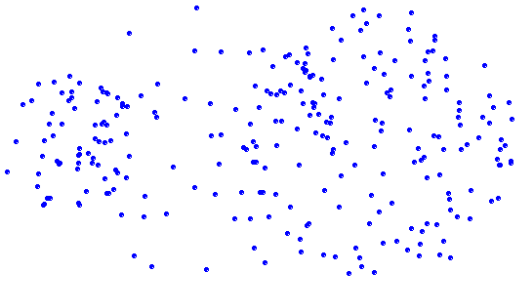
Strength of MIN

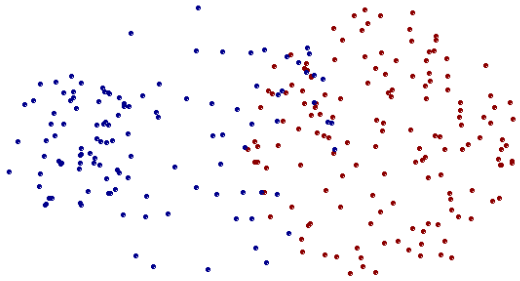




- **Original Points**
- **Two Clusters**
- **Can handle non-elliptical shapes**

Limitations of MIN





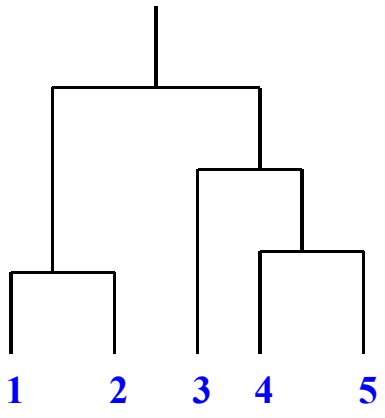
•**Original Points** •**Two Clusters**

• **Sensitive to noise and outliers and perform poorly for overlapping clusters**

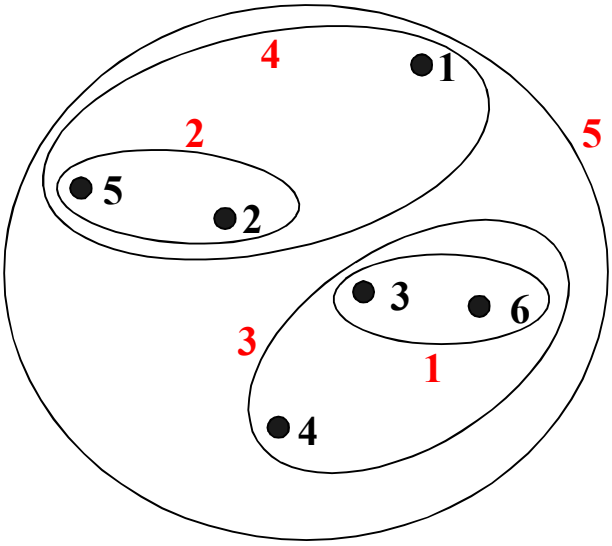
Cluster Similarity: MAX or Complete Linkage

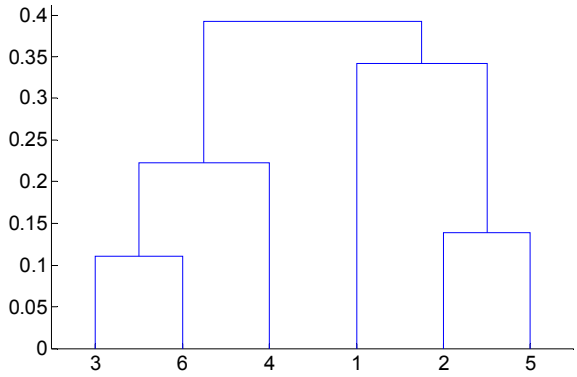
- Distance of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

	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

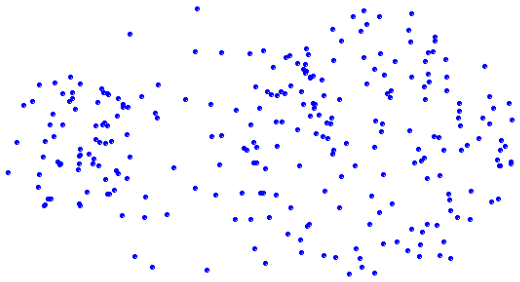




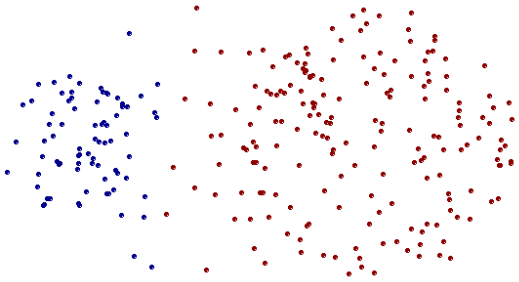
•Nested Clusters

•Dendrogram

Strength of MAX



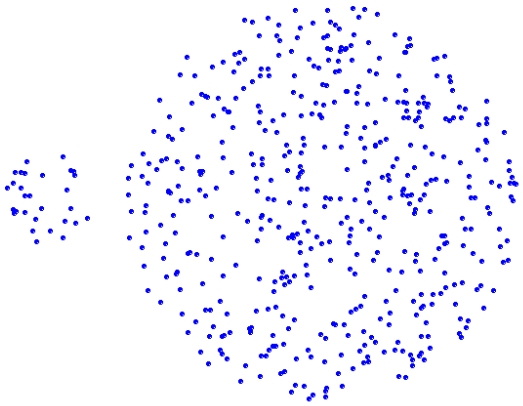
•Original Points

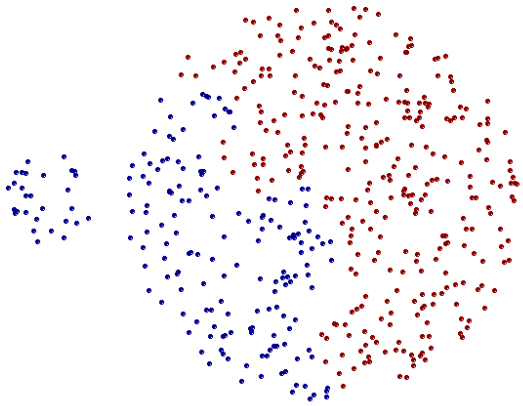


•Two Clusters

• Less susceptible to noise and outliers

Limitations of MAX





- Original Points
- Two Clusters
- Tends to break large clusters

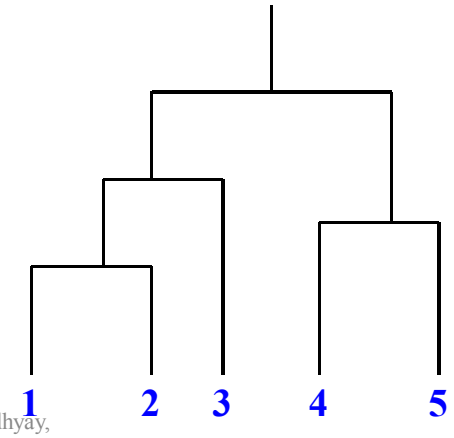
Cluster Distance: Group Average

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

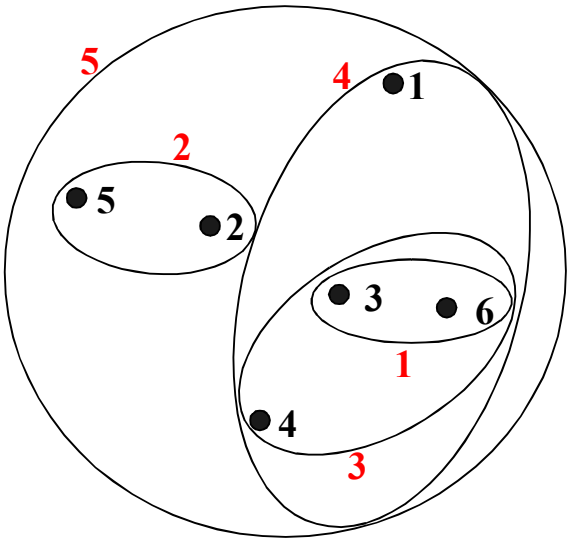
$$\text{Distance}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{Distance}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

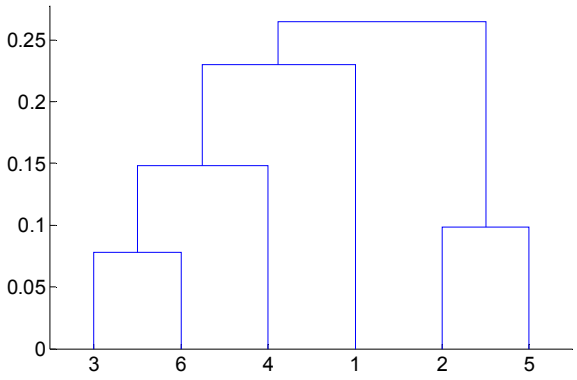
- Need to use average connectivity for scalability since total Distance favors large clusters

	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: Average Linkage



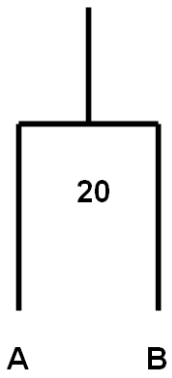


•**Nested Clusters**

•**Dendrogram**

Example of Average Linkage

	A	B	C	D	E
A	0				
B	20	0			
C	60	50	0		
D	100	90	40	0	
E	90	80	50	30	0



New average distance between AB and C is:

C to AB = (60 + 50) / 2 = 55

Distance between D to AB is:

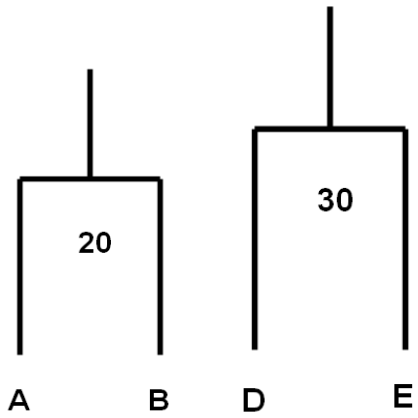
D to AB = (100 + 90) / 2 = 95

Distance between E to AB is:

E to AB = (90 + 80) / 2 = 85

Example of Average Linkage cont 1

	AB	C	D	E
AB	0			
C	55	0		
D	95	40	0	
E	85	50	30	0

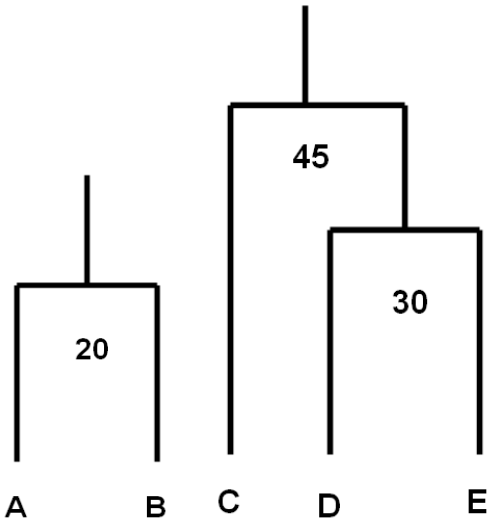


New average distance between AB and DE is:

AB to DE = (95 + 85) / 2 = 90

Example of Average Linkage cont 2

	AB	C	DE
AB	0		
C	55	0	
DE	90	45	0

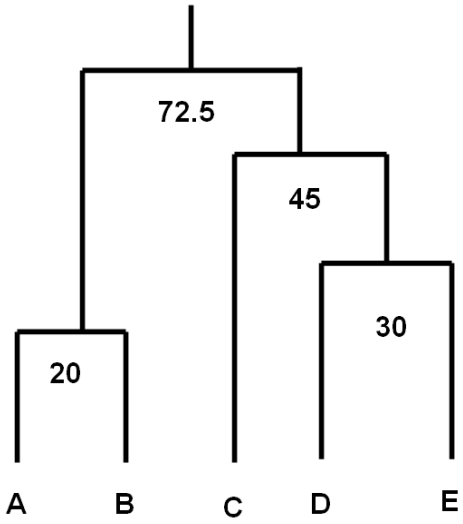


New Average distance between CDE and AB is:

CDE to AB = (90 + 55) / 2 = 72.5

Example of Average Linkage cont 3

	AB	CDE
AB	**0**	
CDE	***72.5***	**0**

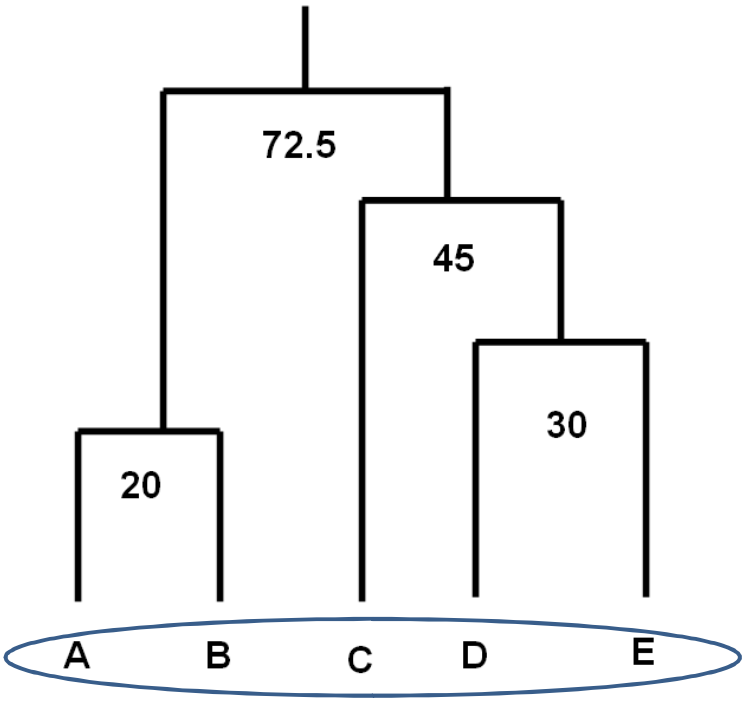


There are only two clusters. so this completes the calculation!

Hierarchical Clustering: Average Linkage

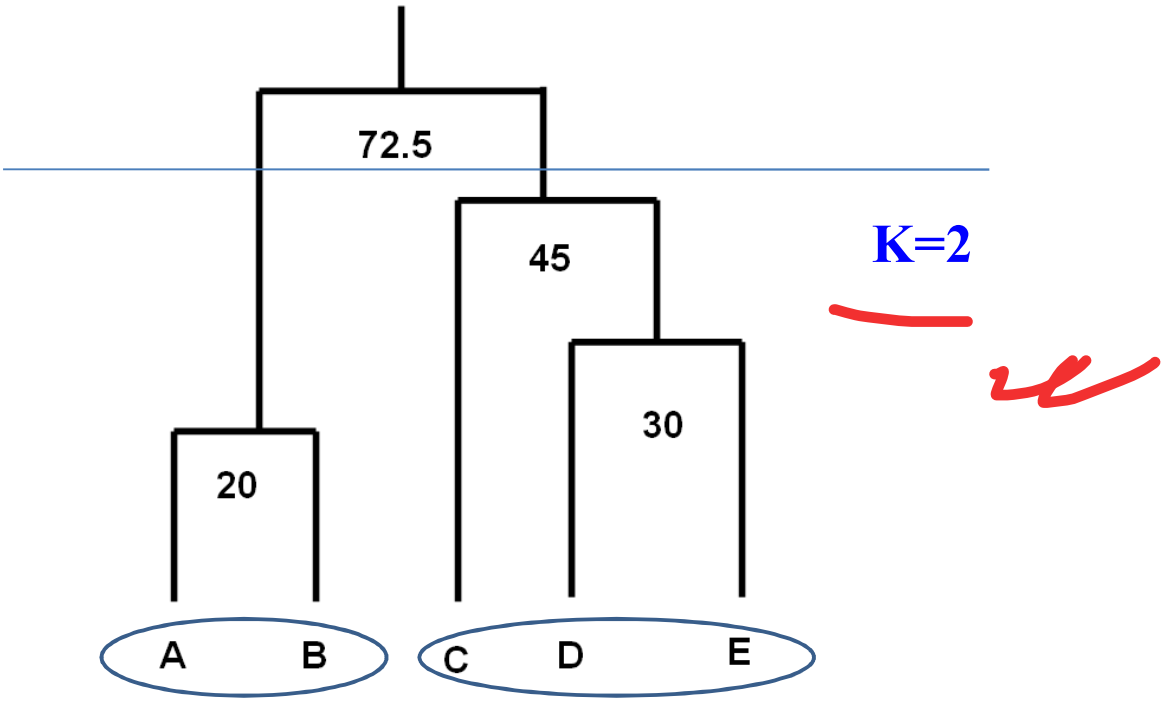
- Compromise between Single and Complete Linkage
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

Getting Clustering from Dedrogram

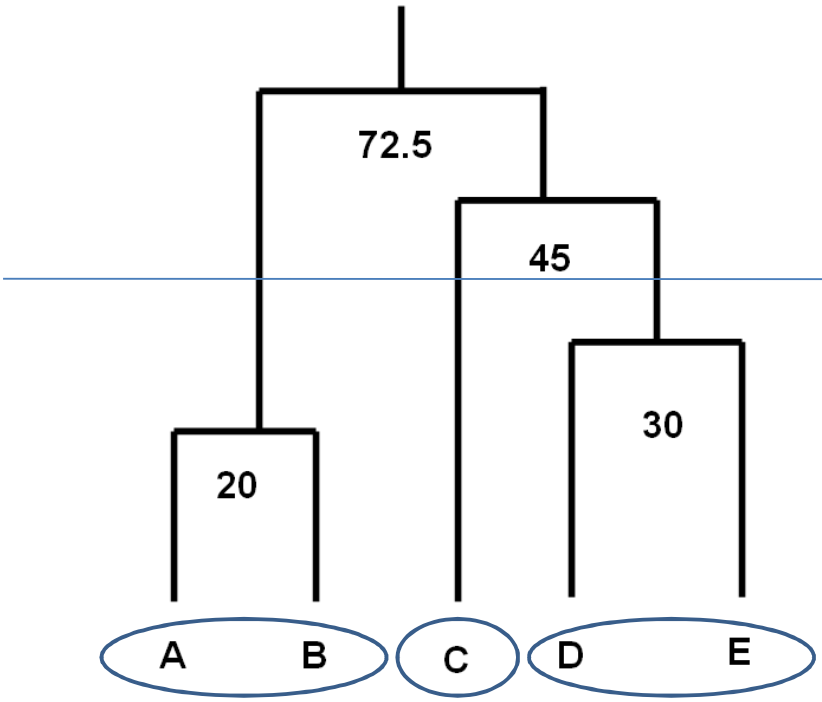


K=1

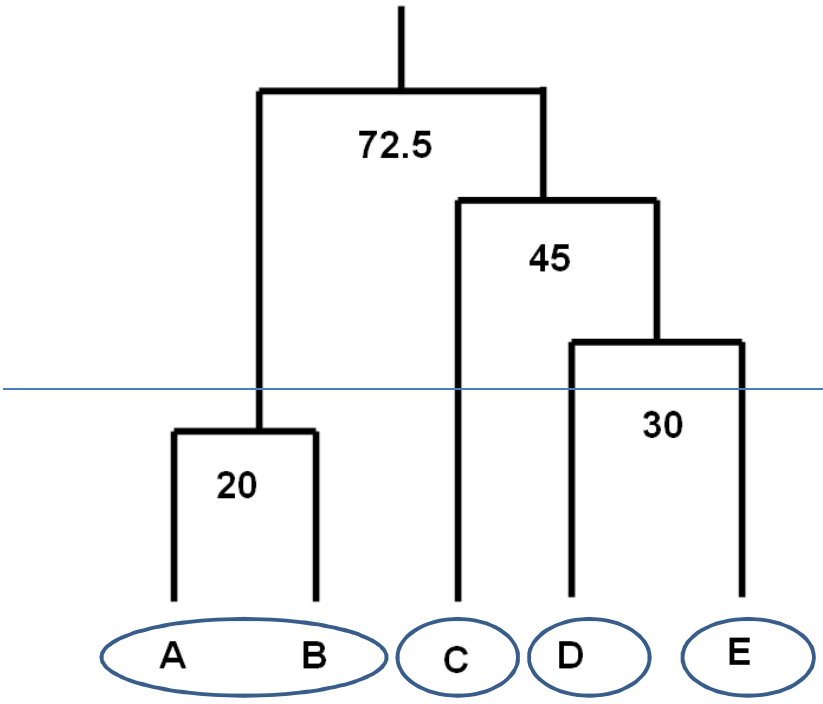
Getting Clustering from Dedrogram



Getting Clustering from Dedrogram

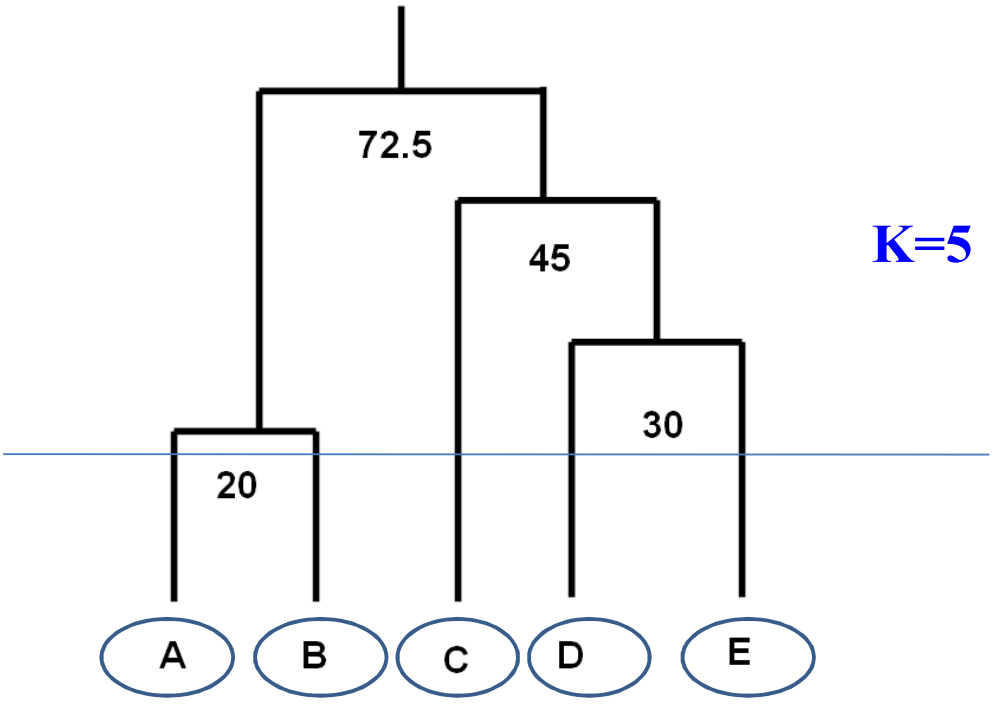


Getting Clustering from Dedrogram



K=4

Getting Clustering from Dedrogram

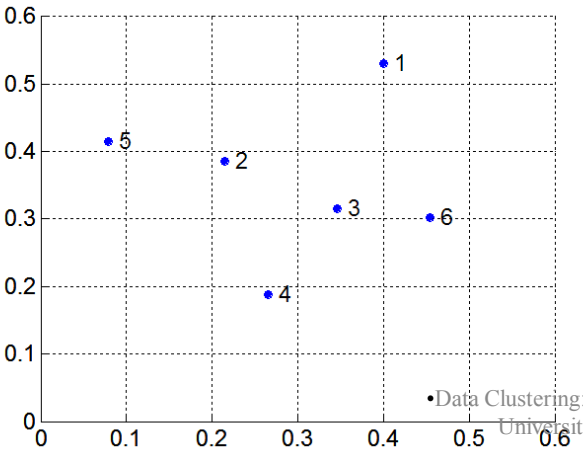


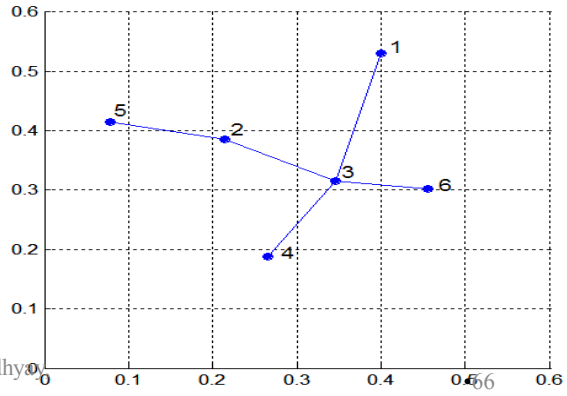
Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone.
- No objective function is directly optimized.
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

MST: Divisive Hierarchical Clustering

- Build MST (Minimum Spanning Tree)
  - Start with a tree that consists of any point
  - In successive steps, look for the closest pair of points (p, q) such that one point (p) is in the current tree but the other (q) is not
  - Add q to the tree and put an edge between p and q





MST: Divisive Hierarchical Clustering

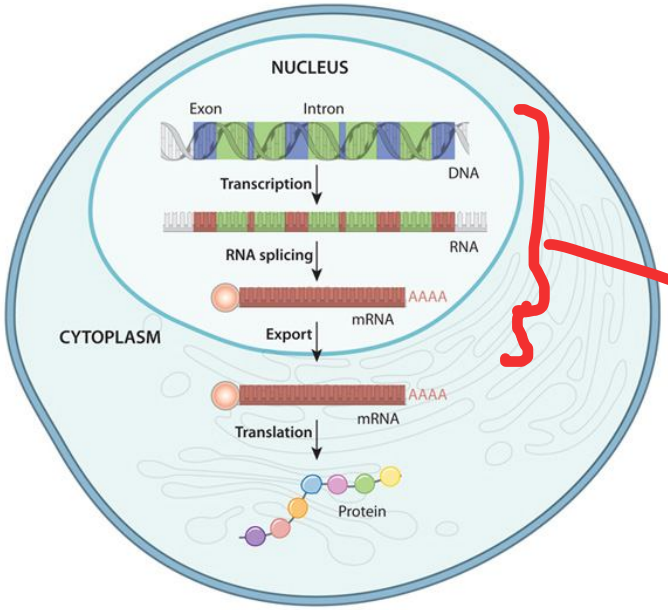
- Use MST for constructing hierarchy of clusters

Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

1: Compute a minimum spanning tree for the proximity graph.
2: **repeat**
3: Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
4: **until** Only singleton clusters remain

An Application of Clustering in Bioinformatics

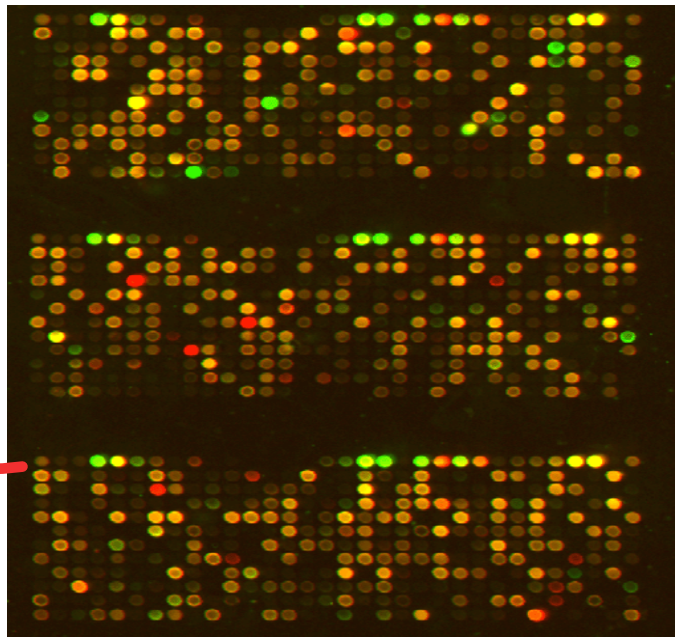
Gene Expression



- **When a gene produces RNA, the gene is expressed.**
- **Expression level** of a gene is also dependent on time
  - **Amount of mRNA** produced varies with time

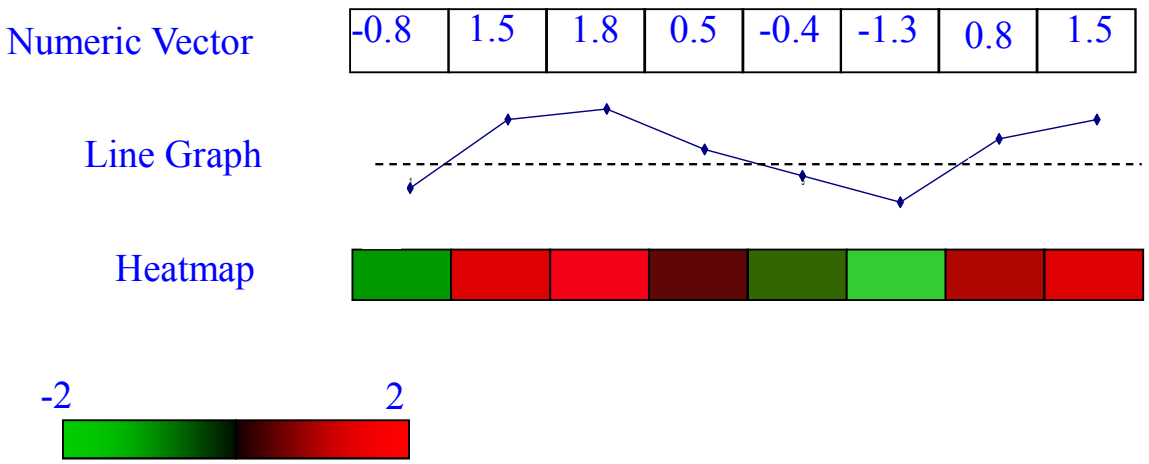
Microarray

- Microarray data set:
  - *G* X *C* matrix *M* ,
    - *G* genes on the rows,
    - *C* conditions/samples on the column
    - M_{ij}=expression level of i*th* gene at j*th* condition.

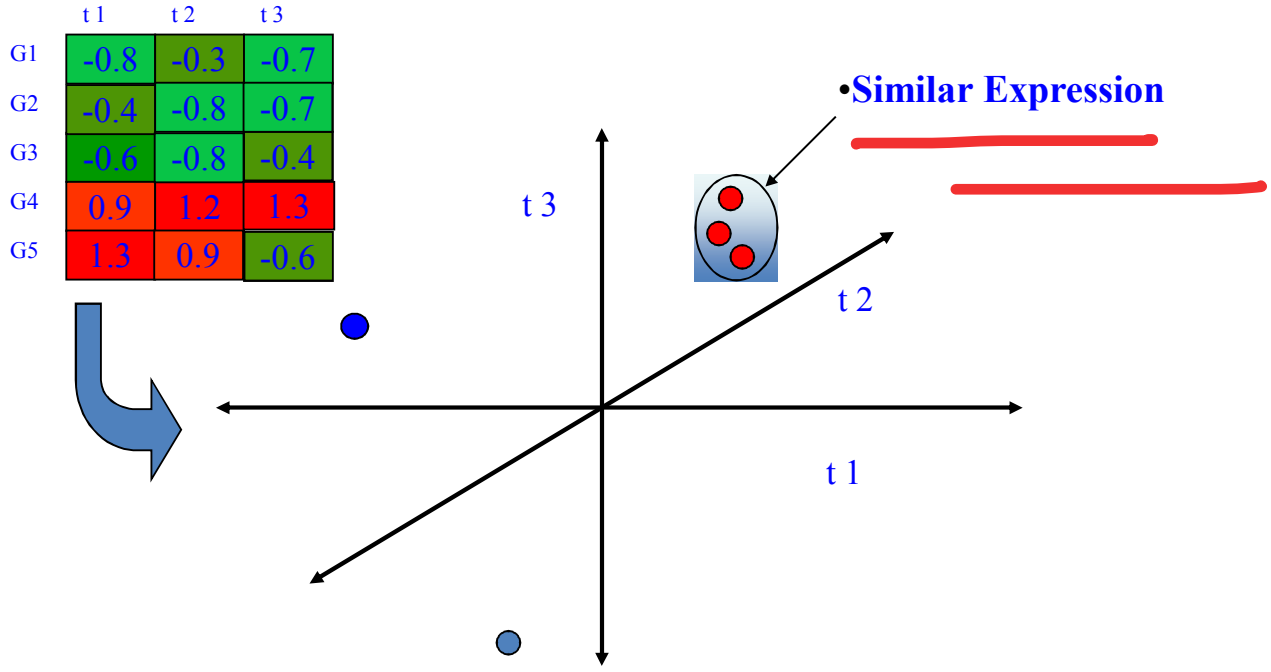


Expression Vectors

Gene Expression Vectors encapsulate the expression of a gene over a set of experimental conditions or sample types.



Expression Vectors As Points in 'Expression Space'



Distance and Similarity

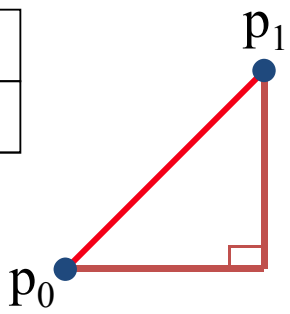
-the ability to calculate a distance (or similarity, it's inverse) between two expression vectors is fundamental to clustering algorithms

-distance between vectors is the basis upon which decisions are made when grouping similar patterns of expression

-selection of a *distance metric* defines the concept of distance

•**Distance: a measure of similarity between gene expression**.

	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
Gene A	x_{1A}	x_{2A}	x_{3A}	x_{4A}	x_{5A}	x_{6A}
Gene B	x_{1B}	x_{2B}	x_{3B}	x_{4B}	x_{5B}	x_{6B}



Some distances:

Euclidean: $\sqrt{\sum_{i=1}^{6}(x_{iA} - x_{iB})^2}$

Manhattan: $\sum_{i=1}^{6}|x_{iA} - x_{iB}|$

Pearson correlation

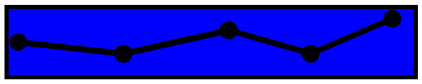
Potential Microarray Applications

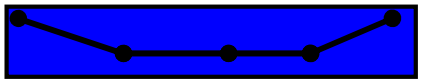
- Drug discovery / toxicology studies
- Mutation/polymorphism detection
- Differing expression of genes over:
  - Time
  - Tissues
  - Disease States
- Sub-typing complex genetic diseases

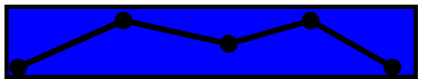
Popular Methods of Clustering of Gene Expression Data

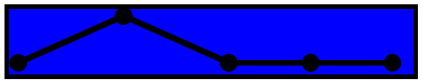
- Hierarchical methods
  - Single link, average link, complete link
    - dendrogram
- K-means Clustering
- GA based Clustering

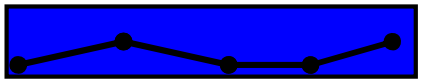
Hierarchical Clustering

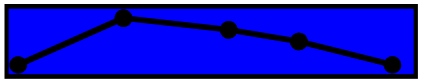
•Gene 1 

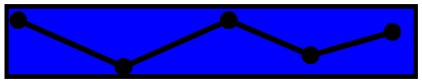
•Gene 2 

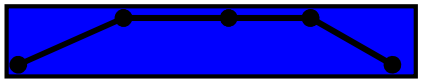
•Gene 3 

•Gene 4 

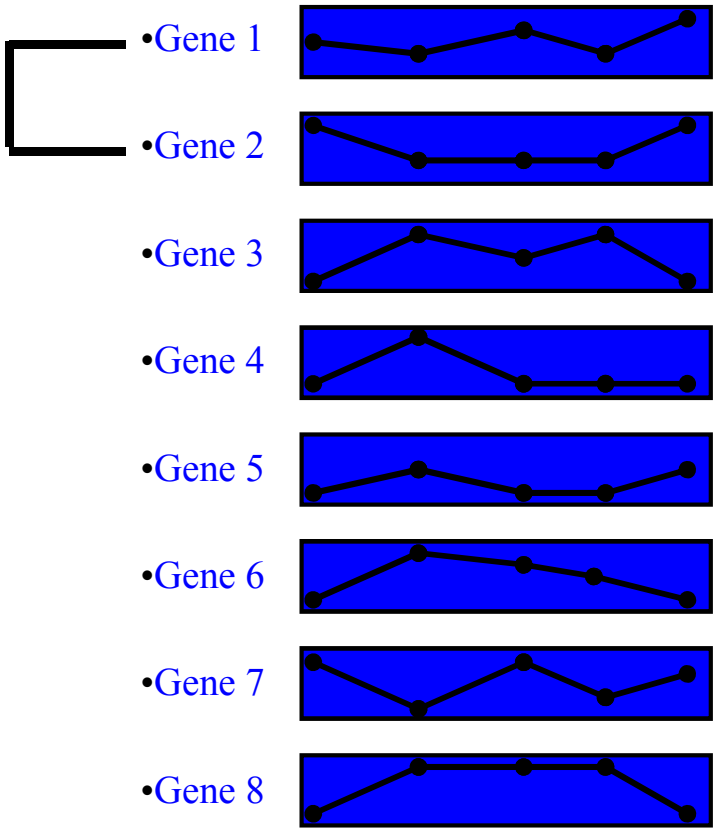
•Gene 5 

•Gene 6 

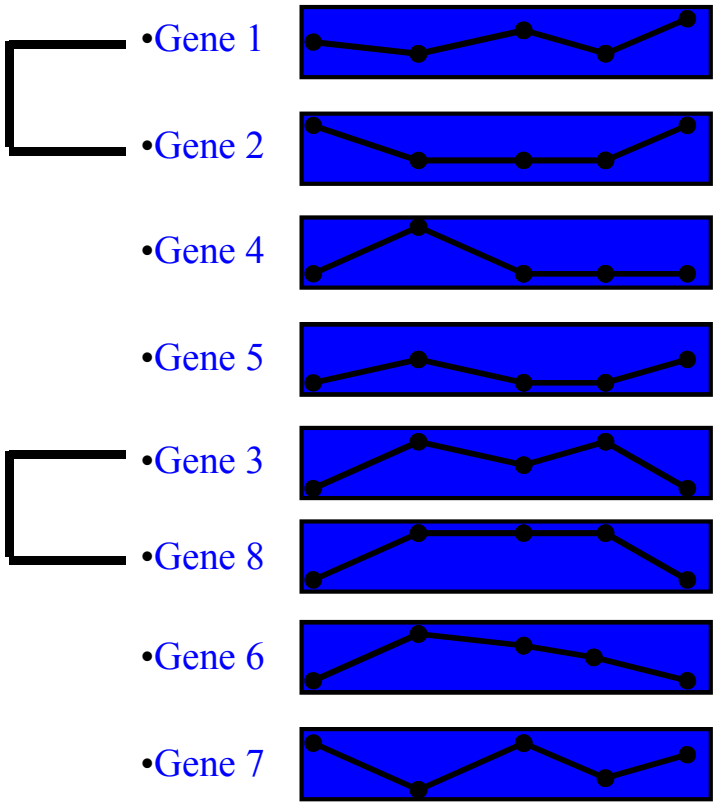
•Gene 7 

•Gene 8 

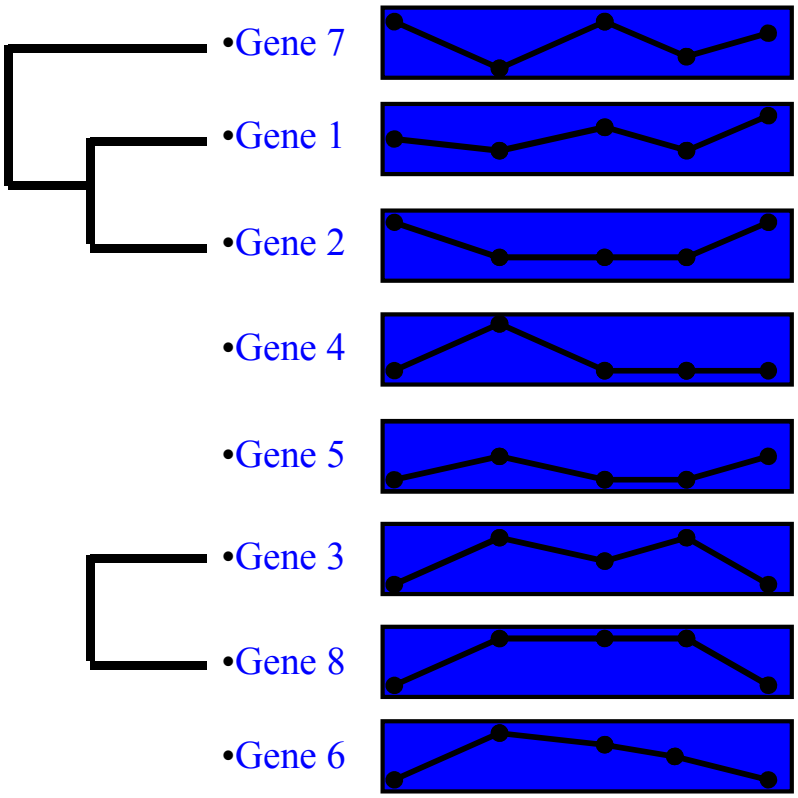
Hierarchical Clustering



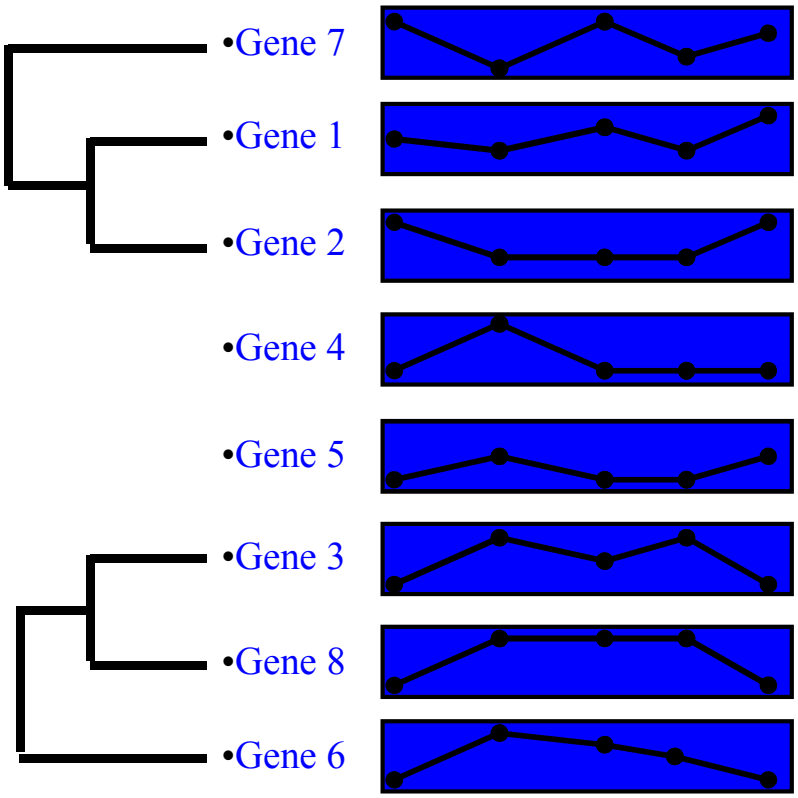
Hierarchical Clustering



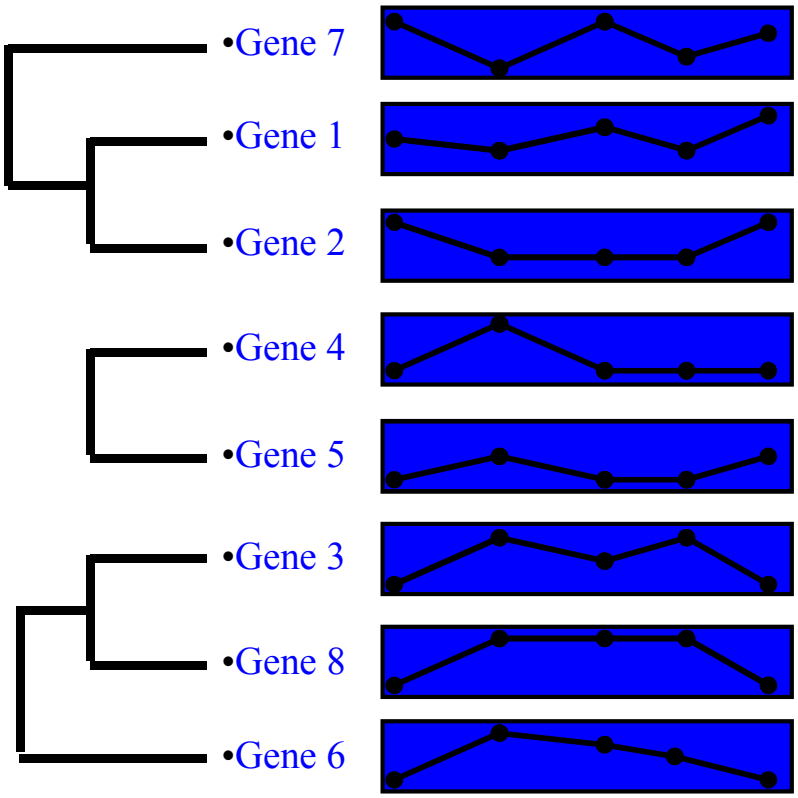
Hierarchical Clustering



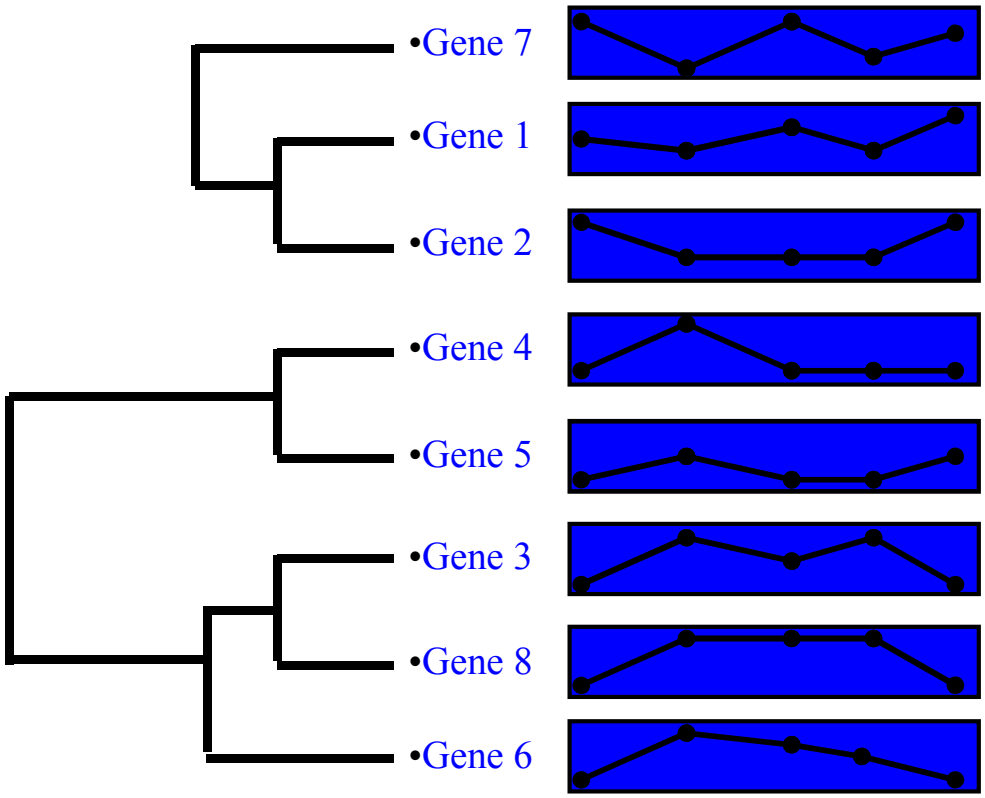
Hierarchical Clustering



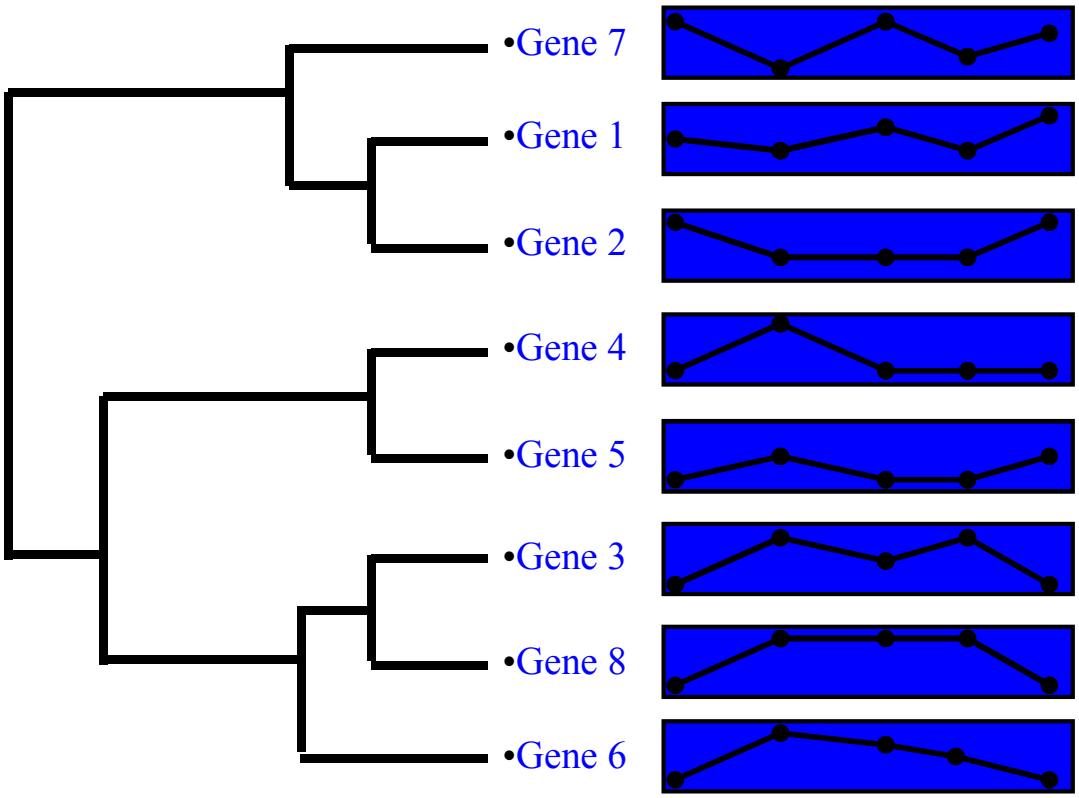
Hierarchical Clustering



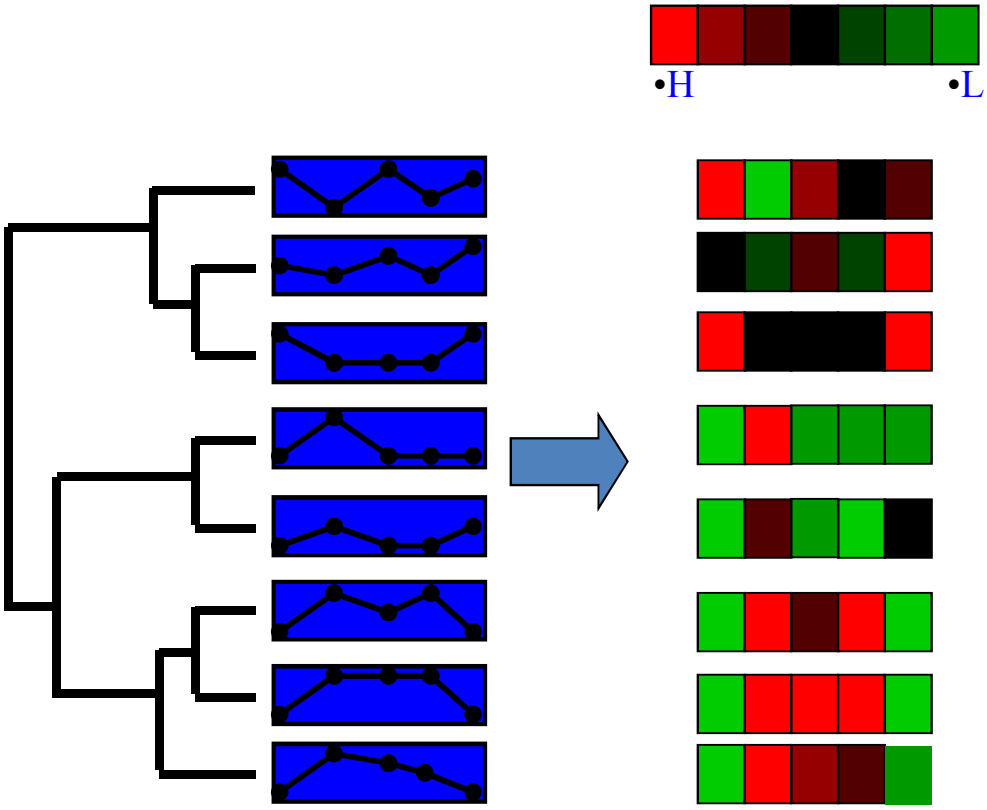
Hierarchical Clustering



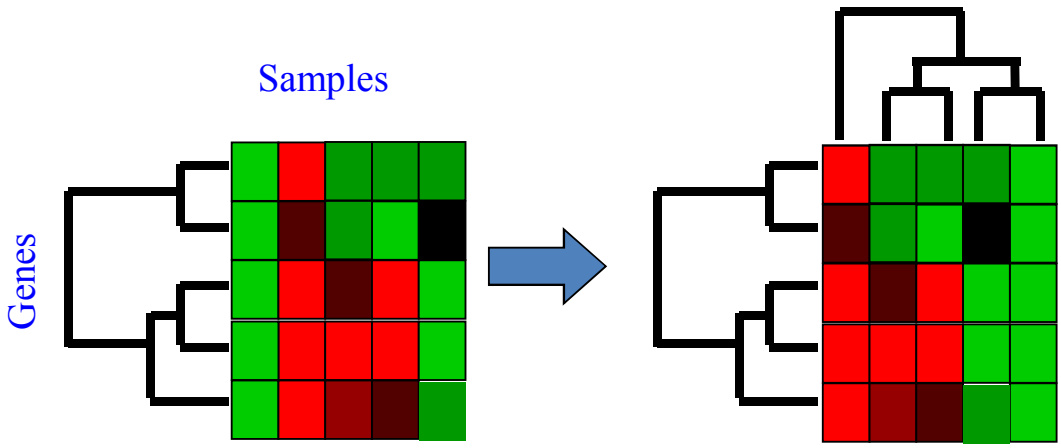
Hierarchical Clustering



Hierarchical Clustering



Hierarchical Clustering



Hierarchical Clustering

- Pros:
  - Easy to understand and implement
- Cons:
  - Real genes probably do not have a hierarchical organization

Experimental results

Data Sets	No. of genes	No. of time points	No. of clusters
Yeast Sporulation	6118	7	7
Human Fibroblasts Serum	517	13	10

Experimental results (Cont.)

- Performance metric: *Silhouette index*
  - Silhouette width of a point is defined as:

$$s = \frac{b - a}{\max\{a, b\}}.$$

  - **a**: the average distance of the point from the other points of the cluster to which the point is assigned.
  - **b**: the minimum of the average distances of the point from the points of the other clusters.

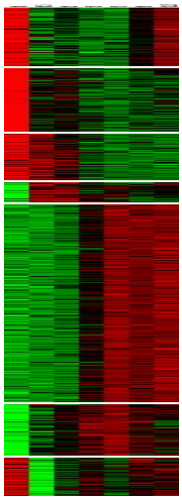
- Silhouette index is the average silhouette width of all the data points (genes). It ranges between -1 and 1, and larger value indicates better solution.

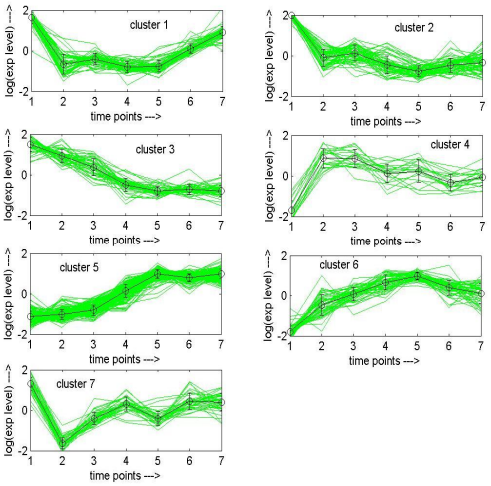
Experimental results (Cont.)

Algorithm	Data Set	
	Sporulation	Serum
K-means	0.5783	0.3298
Hierarchical Average linkage	0.5007	0.2977
Single objective GA with objective function J	0.5833	0.3498
Multiobjective GA with objective functions J and I_{XB} indices	0.6478	0.4204

Silhouette index values for different algorithms on Sporulation and Serum data sets

Visualizing clustering results



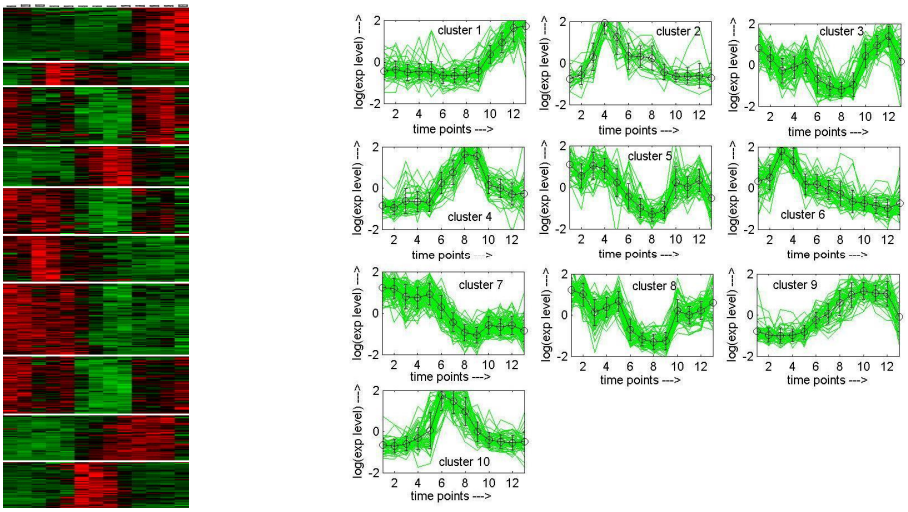


S. Bandyopadhyay, A. Mukhopadhyay, U. Maulik, "An Improved Algorithm for Clustering Gene Expression Data", *Bioionf.*, vol. 23, no. 21, pp. 2859-2865, 2007.

(a) (b)

Sporulation data clustered using multiobjective clustering (7 clusters): (a) Eisen plot, (b) Cluster profile plots.

Visualizing clustering results (Cont.)



(a) (b)

Serum data clustered using multiobjective clustering (10 clusters): (a) Eisen plot, (b) Cluster profile plots.

References

- K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

- S. Bandyopadhyay, **A. Mukhopadhyay** and U. Maulik, “An Improved Algorithm for Clustering Gene Expression Data”, Bioinformatics, Vol. 23, No. 21, pp. 2859-2865, 2007.

- **A. Mukhopadhyay**, U Maulik and S. Bandyopadhyay, “Multi-objective Genetic Algorithm based Fuzzy Clustering of Categorical Attributes”, IEEE Transactions on Evolutionary Computation, Vol. 13, No. 5, pp. 991-1005, 2009.

- U. Maulik, S. Bandyopadhyay and **A. Mukhopadhyay**, "Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics", Springer, Heidelberg-Berlin, ISBN 978-3-642-16614-3, 2011.