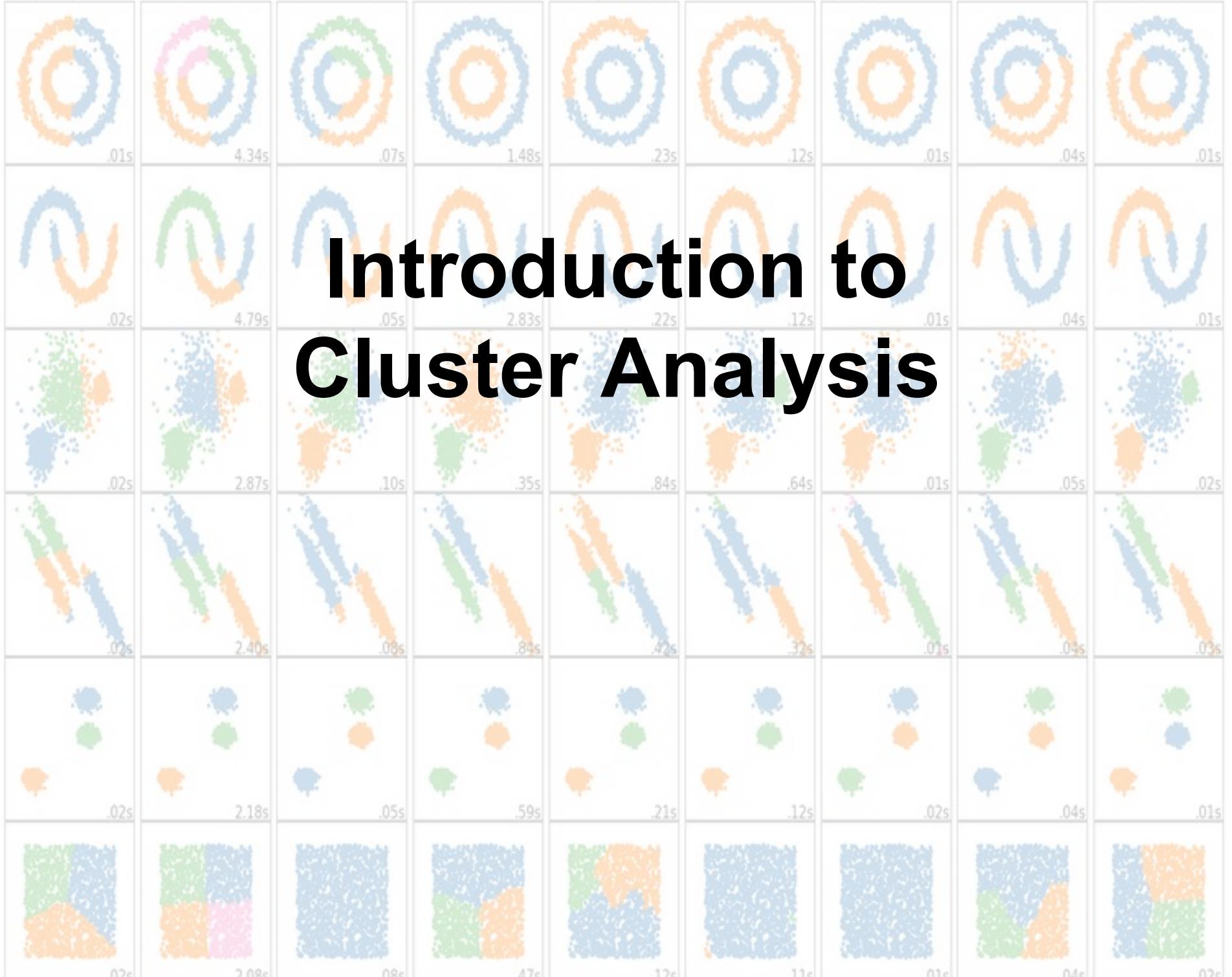


MiniBatchKMeans AffinityPropagation MeanShift SpectralClustering Ward AgglomerativeClustering DBSCAN Birch GaussianMixture



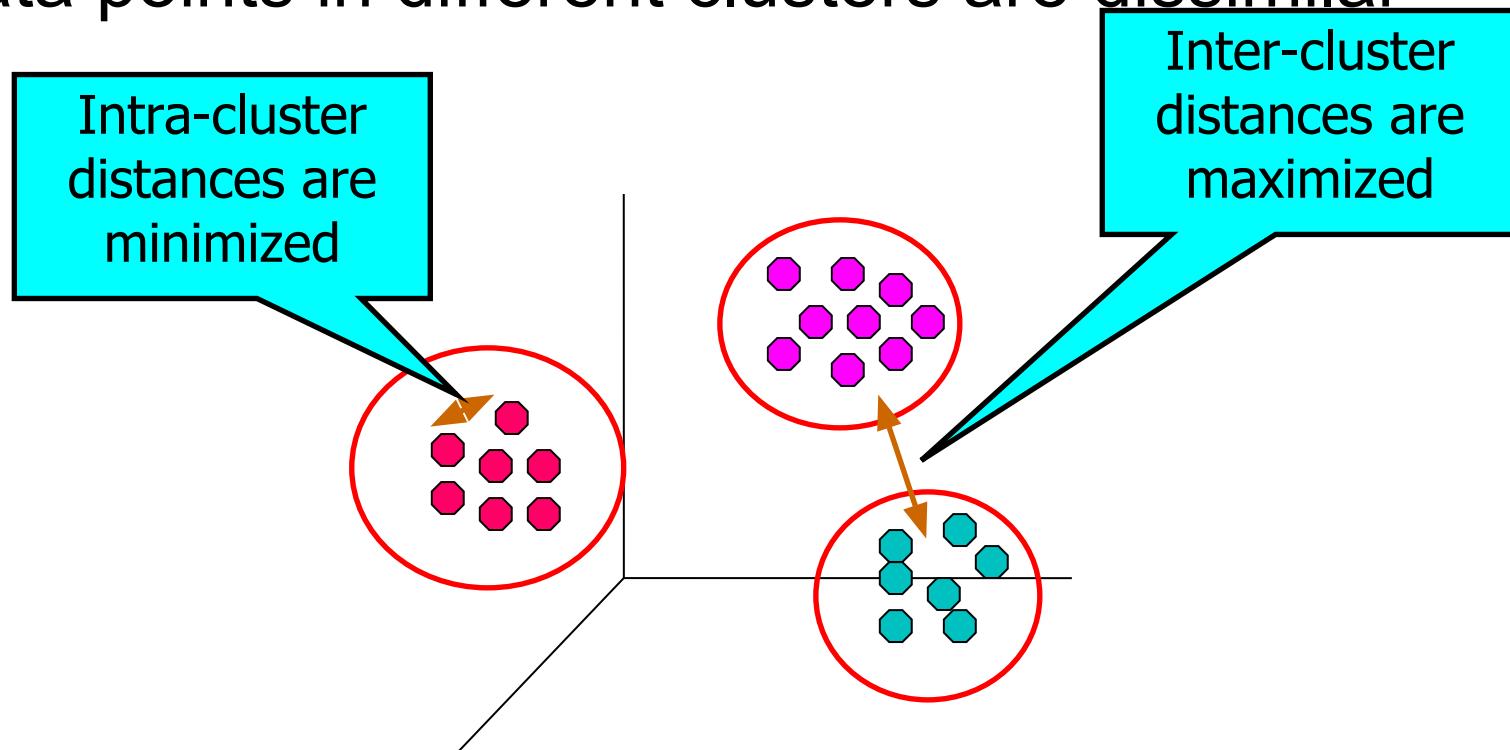
# Introduction to Cluster Analysis

# Outline

- Background
  - Intro
  - Workflow
  - Similarity metrics
- Clustering algorithms
  - Hierarchical
  - K-means
  - Density-based
- Cluster evaluation
  - External
  - Internal

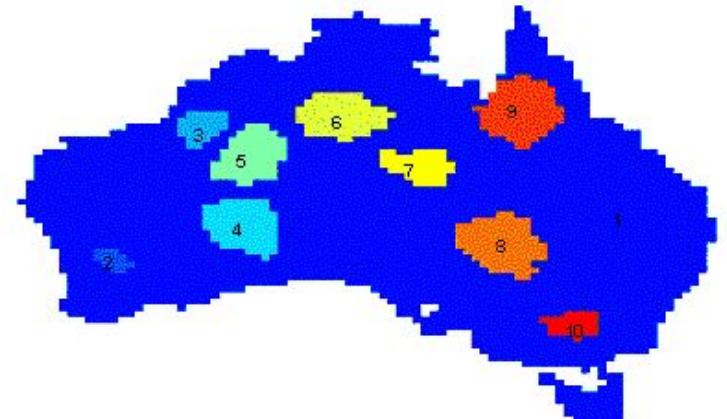
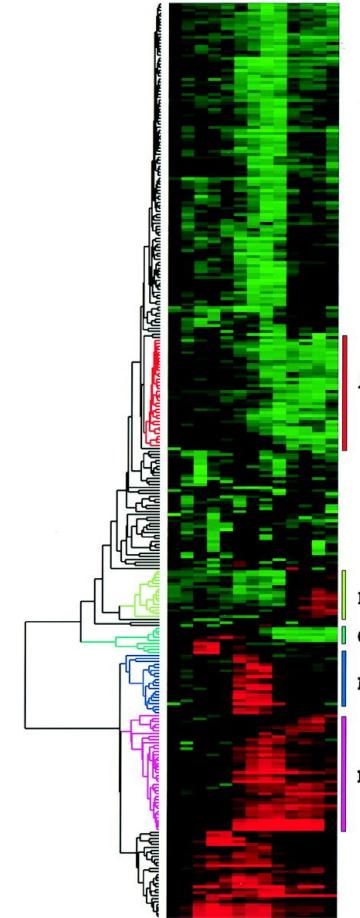
# Cluster Analysis

- Data mining tool(s) for dividing a multivariate dataset into (meaningful, useful) groups
- Good clustering:
  - Data points in one cluster are highly similar
  - Data points in different clusters are dissimilar



# Applications

- Gain understanding
  - Groups of genes/proteins with similar function (from nucleotide or amino acid sequence data)
  - Groups of cells with similar expression patterns (from RNAseq data)
- Summarize
  - Reduce the size of a large dataset



Clustering precipitation  
in Australia

# Cluster analysis is not...

## Simple segmentation

i.e., Dividing students into different registration groups alphabetically, by last name

Although, some work in graph partitioning and more complex segmentation is related to clustering

## The results of a query

Groupings are a result of an external specification

## Supervised classification

Supervised classification has class label information

Clustering can be called **unsupervised** classification: labels derived from data

## Association Analysis

Finding connections between items in datasets

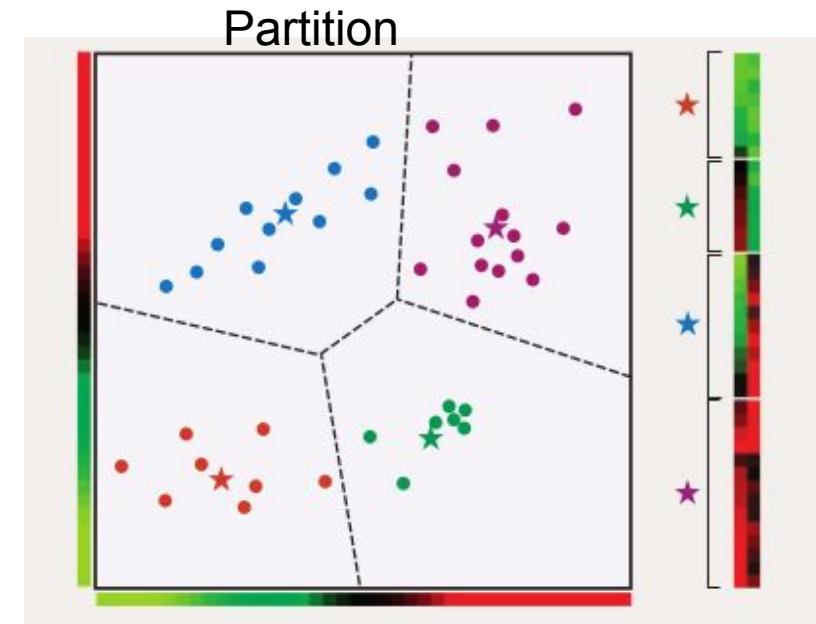
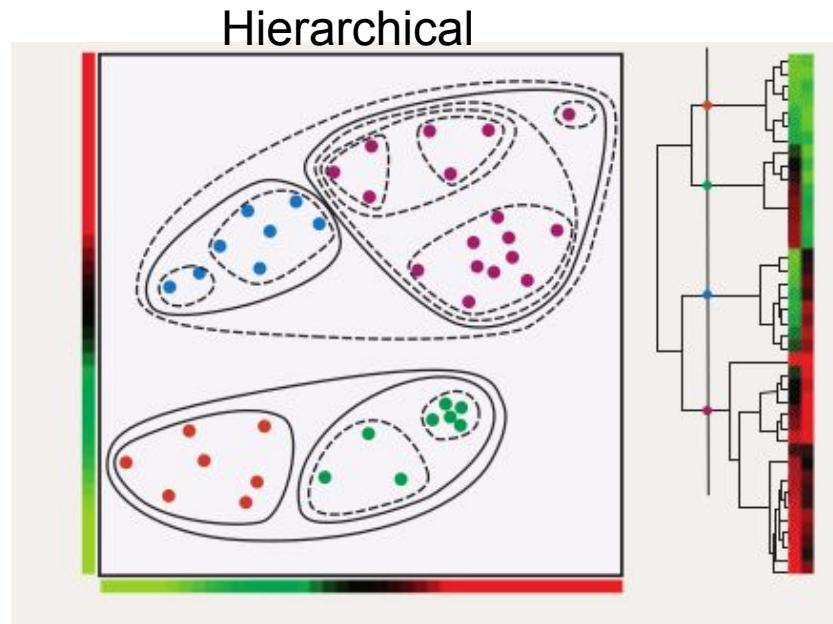
# Cluster evaluation has an element of subjectivity



(a) Original points.

# Traditional types of clusterings

- A clustering is a set of clusters
- Clusters can be:
  - **Hierarchical**: data are in nested clusters, organized in a hierarchical tree
  - **Partition**: data in non-overlapping subsets. One data object is in one subset.



# Other distinctions between clusters

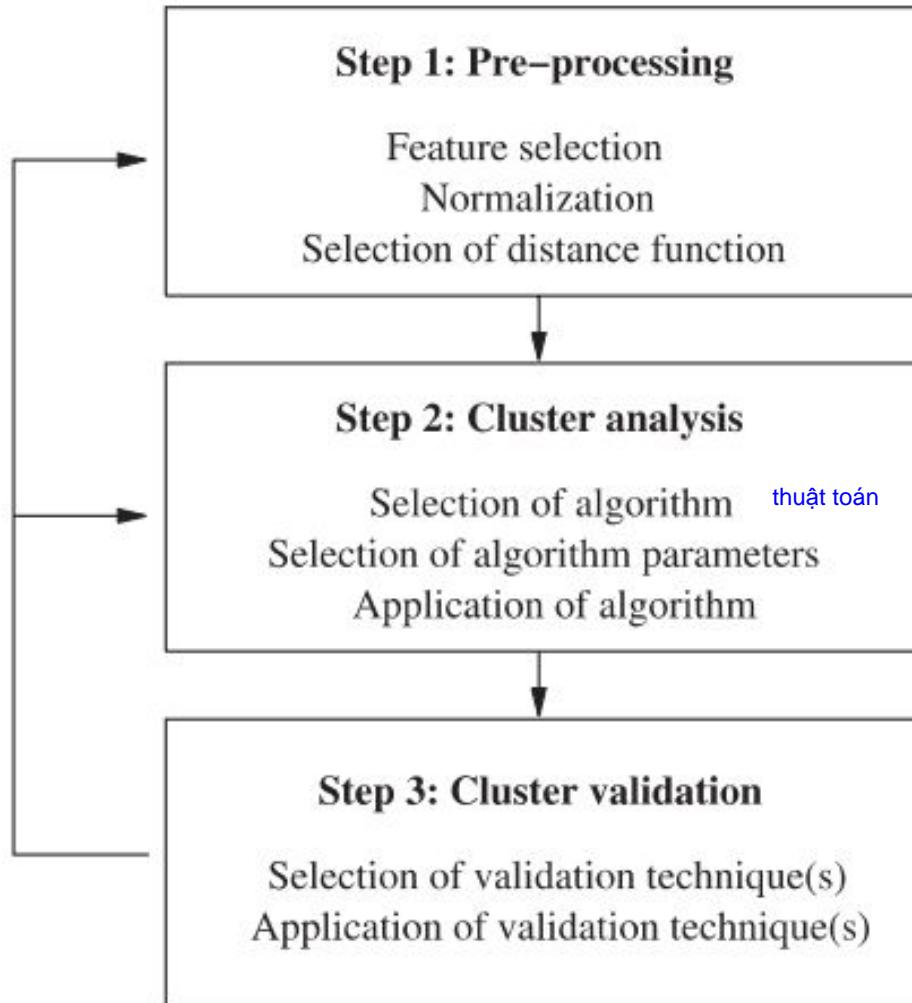
- **Exclusive vs non-exclusive**
  - Exclusive: points belong to one cluster
  - Non-exclusive: points can belong to multiple
- **Fuzzy vs non-fuzzy**
  - In fuzzy clustering, a point belongs to every cluster with some weight (0 to 1)
  - Weights must sum to 1
  - Similar to probabilistic clustering
- **Partial vs complete**
  - Partial: only some of the data is clustered (can exclude outliers)
- **Heterogenous vs homogeneous**
  - Degree to which cluster size, shape, and density can vary

# Why is cluster analysis hard?

- Clustering in two dimensions looks easy!
- Clustering small amounts of data looks easy
- In most cases, looks are **not** deceiving
- However, many applications involve more than 2 dimensions (i.e., human gene expression dataset has  $>10,000$  dimensions)
- **High dimensional spaces look different:** Almost all pairs of points are at about the same distance



# Typical workflow for cluster analysis



# Similarity (aka distance) metrics

**Table 1 Gene expression similarity measures**

Manhattan distance

(city-block distance, L1 norm)

$$d_{fg} = \sum_c |e_{fc} - e_{gc}|$$

Euclidean distance

(L2 norm)

$$d_{fg} = \sqrt{\sum_c (e_{fc} - e_{gc})^2}$$

Mahalanobis distance

$$d_{fg} = (\mathbf{e}_f - \mathbf{e}_g)' \boldsymbol{\Sigma}^{-1} (\mathbf{e}_f - \mathbf{e}_g), \text{ where } \boldsymbol{\Sigma} \text{ is the (full or within-cluster) covariance matrix of the data}$$

Pearson correlation

(centered correlation)

$$d_{fg} = 1 - r_{fg}, \text{ with } r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2 \sum_c (e_{gc} - \bar{e}_g)^2}}$$

Uncentered correlation

(angular separation, cosine angle)

$$d_{fg} = 1 - r_{fg}, \text{ with } r_{fg} = \frac{\sum_c e_{fc} e_{gc}}{\sqrt{\sum_c e_{fc}^2 \sum_c e_{gc}^2}}$$

Spellman rank correlation

As Pearson correlation, but replace  $e_{gc}$  with the rank of  $e_{gc}$  within the expression values of gene  $g$  across all conditions  $c = 1 \dots C$

Absolute or squared correlation

$$d_{fg} = 1 - |r_{fg}| \text{ or } d_{fg} = 1 - r_{fg}^2$$

$d_{fg}$ , distance between expression patterns for genes  $f$  and  $g$ .  $e_{gc}$ , expression level of gene  $g$  under condition  $c$ .

# Outline

- Background
  - Intro
  - Workflow
  - Similarity metrics
- **Clustering algorithms**
  - Hierarchical
  - K-means
  - Density-based
- Cluster evaluation
  - External
  - Internal

# Hierarchical clustering

Produces nested clusters

Can be visualized as a

**dendrogram**

Can be either:

- **Agglomerative** (bottom up):

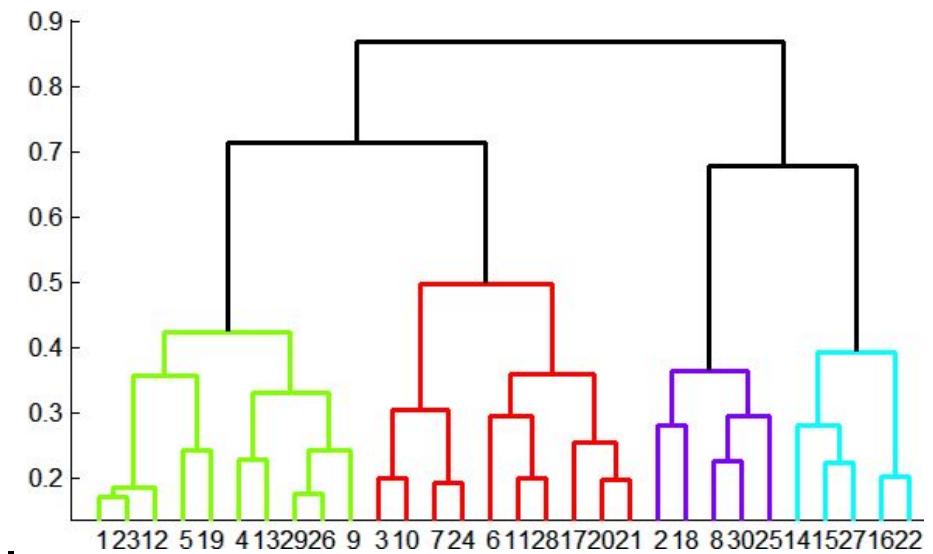
Initially, each point is a cluster

Repeatedly combine the two

“nearest” clusters into one

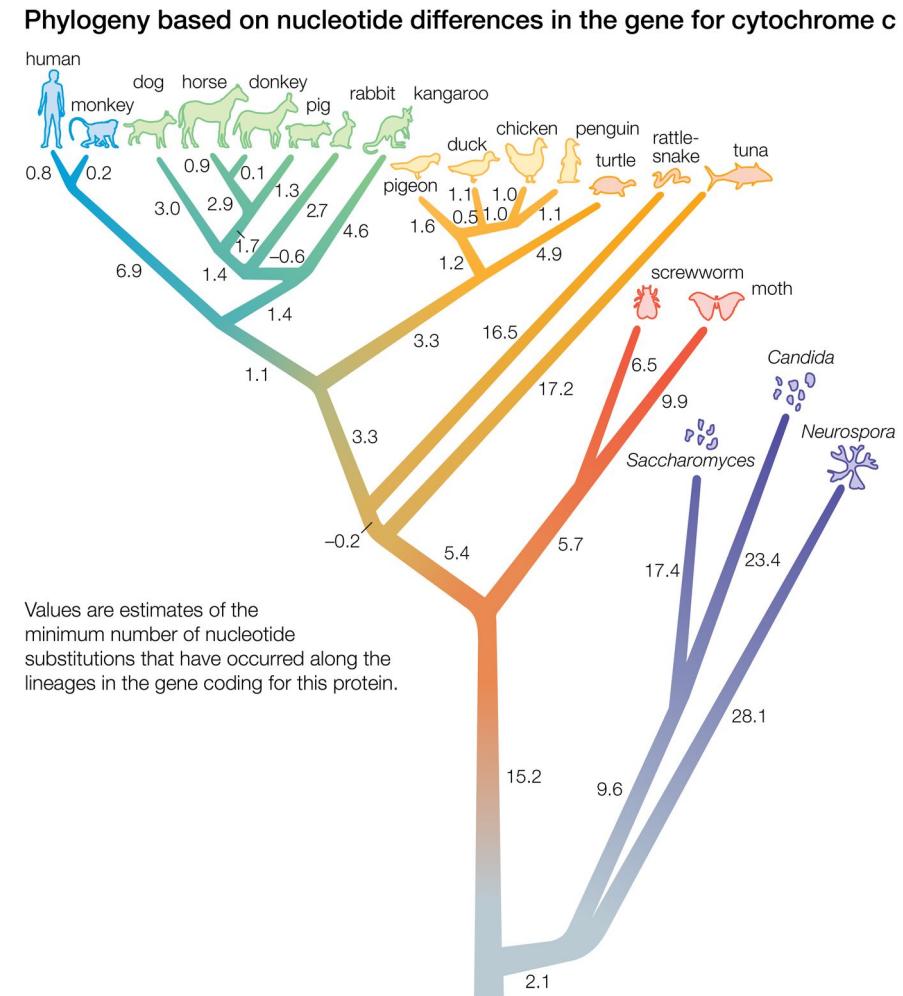
- **Divisive** (top down): phân bậc

Start with one cluster and recursively  
split



# Advantages 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
- No random component (clusters will be the same from run to run)
- Clusters may correspond to meaningful taxonomies
  - Especially in biological sciences (e.g., phylogeny reconstruction)

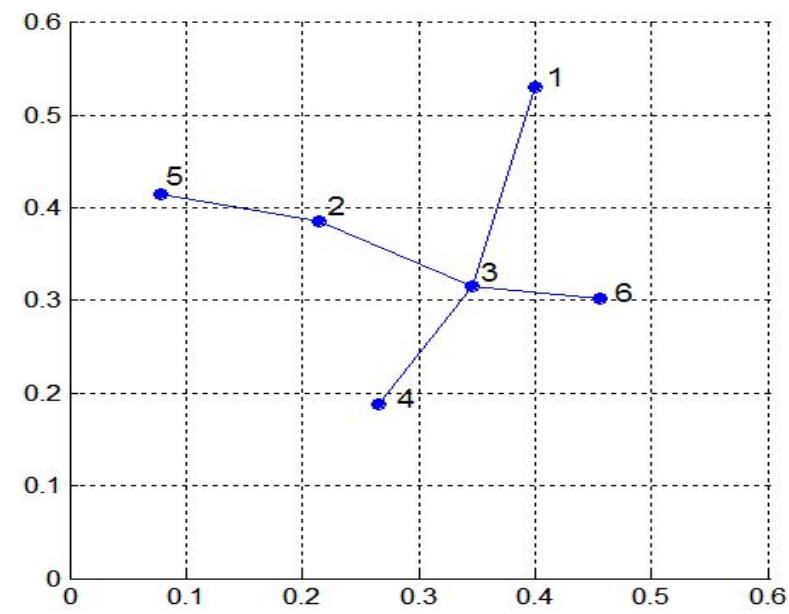
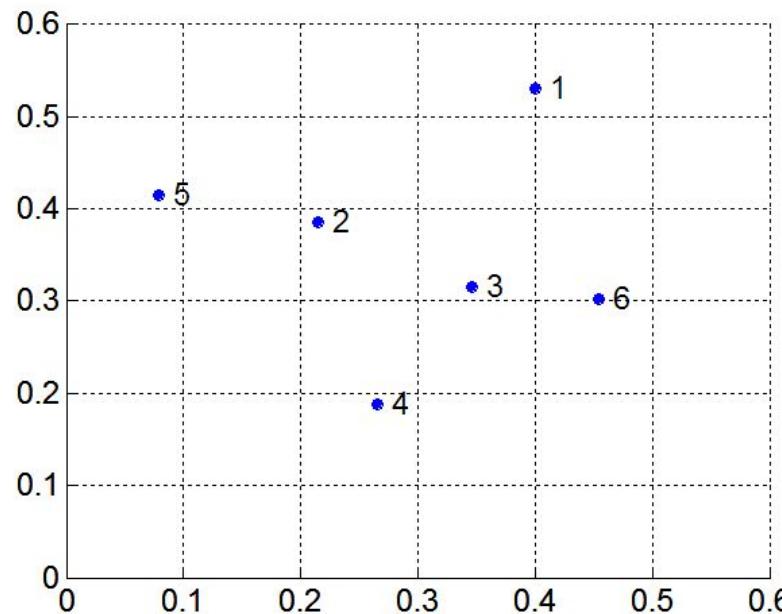


# Agglomerative Clustering Algorithm

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

# Divisive Clustering Algorithm

- Minimum spanning tree (MST)
  - Start with one point
  - In successive steps, look for closest pair of points  $(p, q)$  such that  $p$  is in the tree but  $q$  is not.
  - Add  $q$  to the tree (add edge between  $p$  and  $q$ )

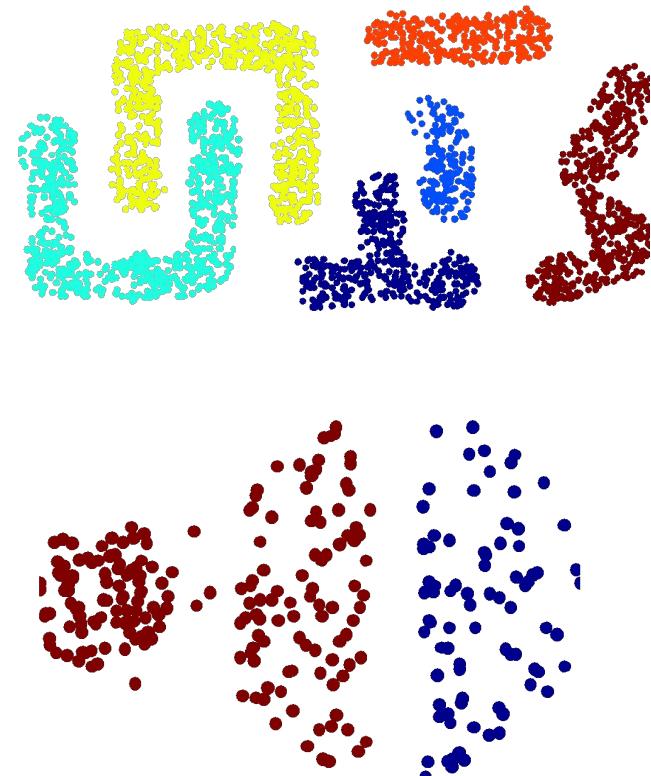
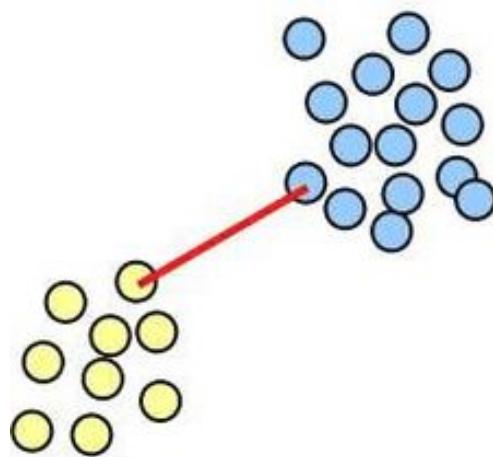


# Linkages

- Linkage: measure of dissimilarity between clusters
- Many methods:
  - Single linkage
  - Complete linkage
  - Average linkage
  - Centroids
  - Ward's method

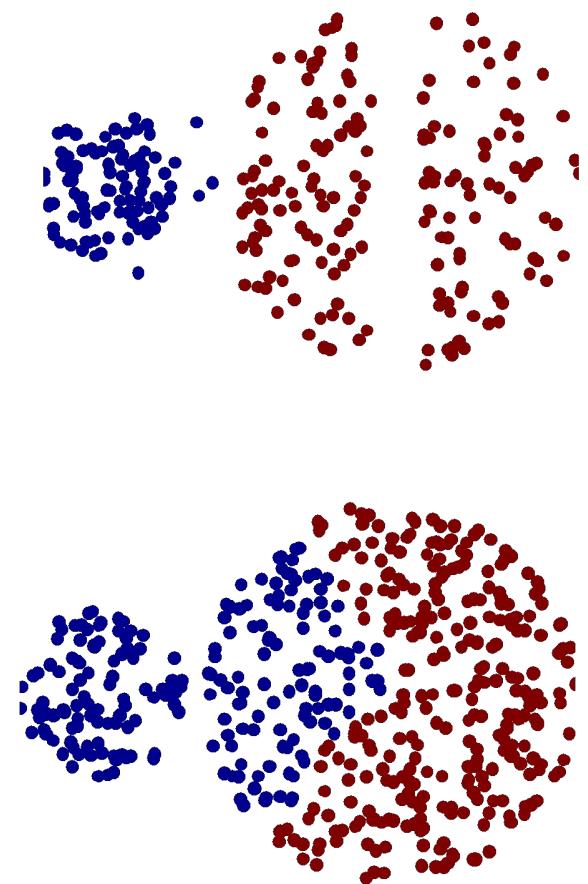
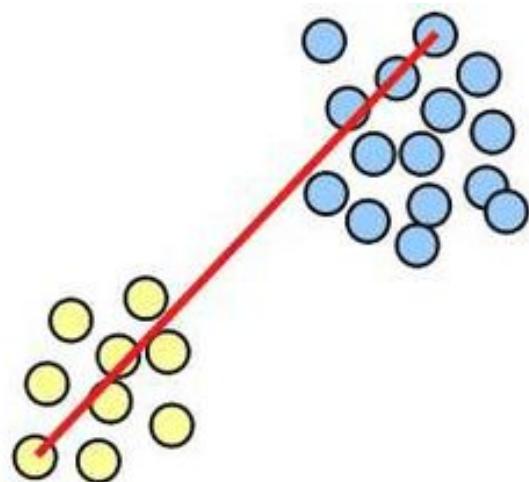
# Single linkage (aka nearest neighbor)

- Proximity of two clusters is based on the two closest points in the different cluster
- Proximity is determined by one pair of points (i.e., one link)
- Can handle non-elliptical shapes
- Sensitive to noise and outliers



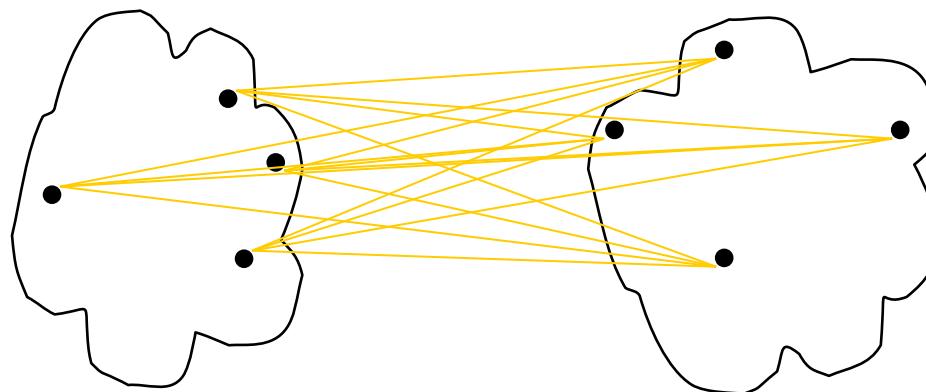
# Complete linkage

- Proximity of two clusters is based on the two most distant points in the different clusters
- Less susceptible to noise and outliers
- May break large clusters
- Biased toward globular clusters



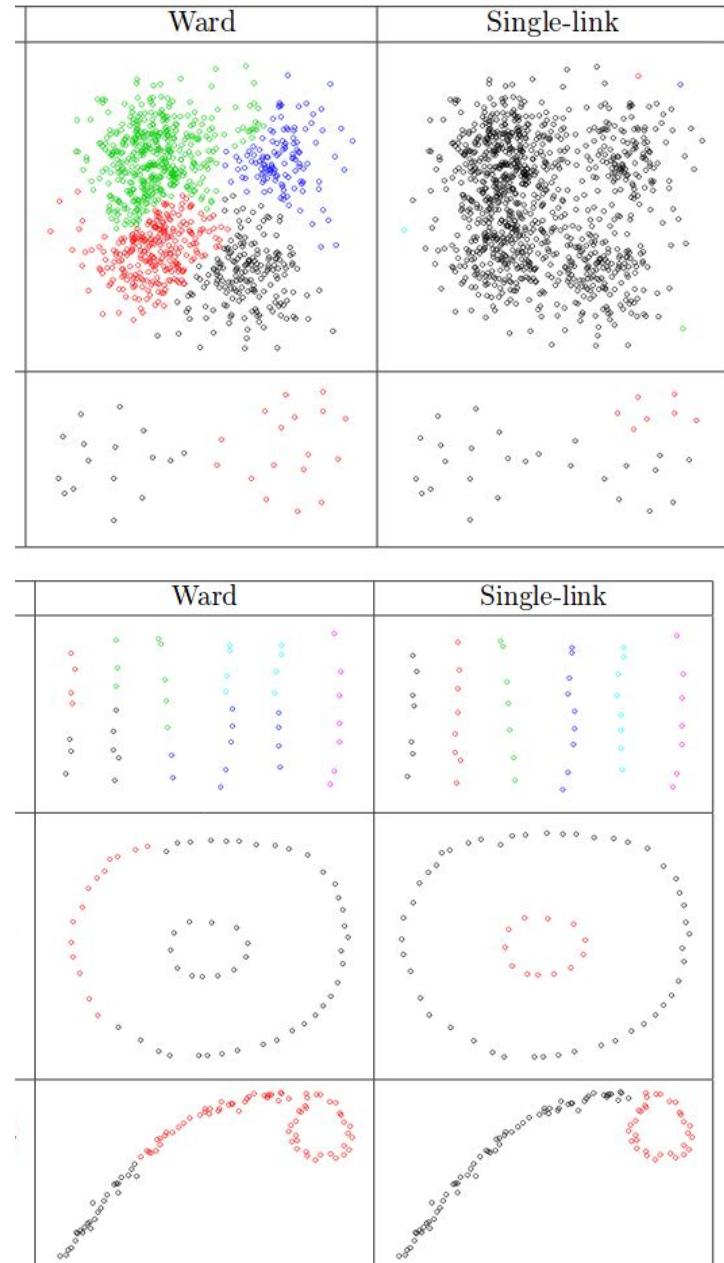
# Average linkage

- Proximity of two clusters is the average of pairwise proximity between points in the clusters
- Less susceptible to noise and outliers
- Biased towards globular clusters



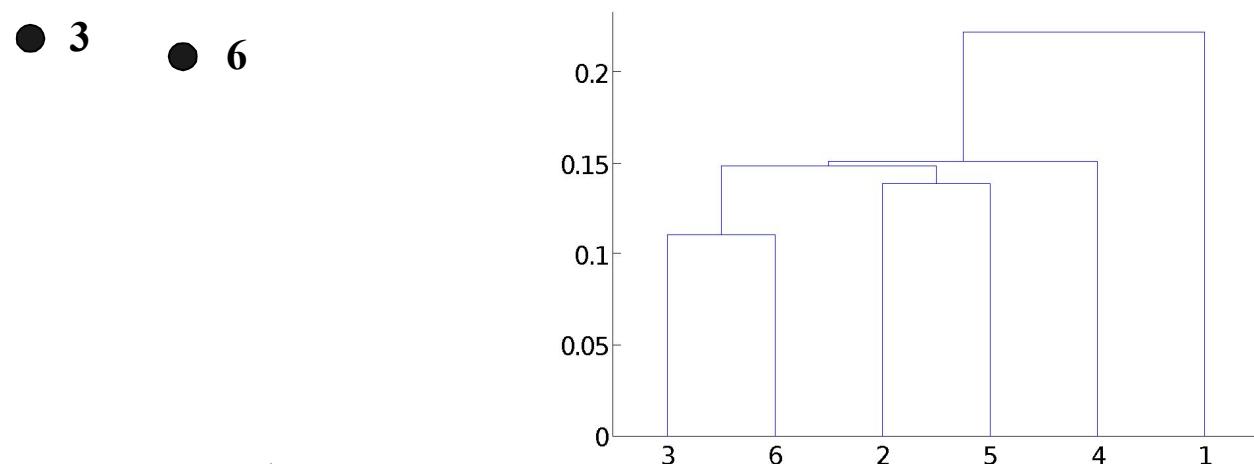
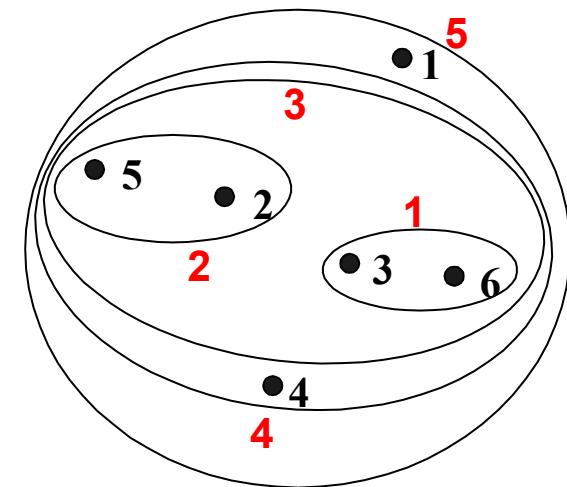
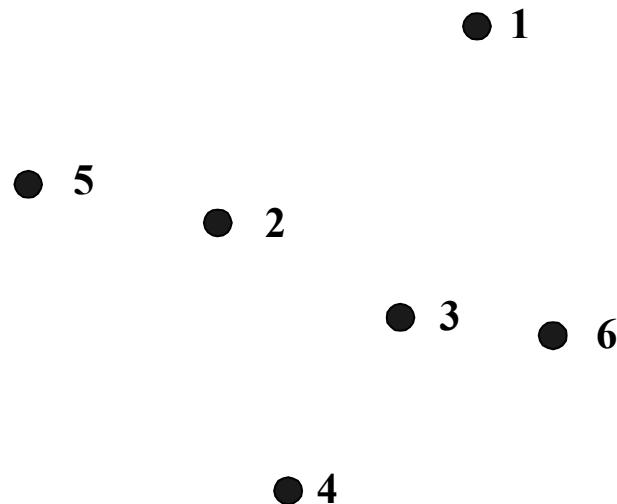
# Ward's method

- Similarity of two clusters is based on the increase in squared error when two clusters are merged
- Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters



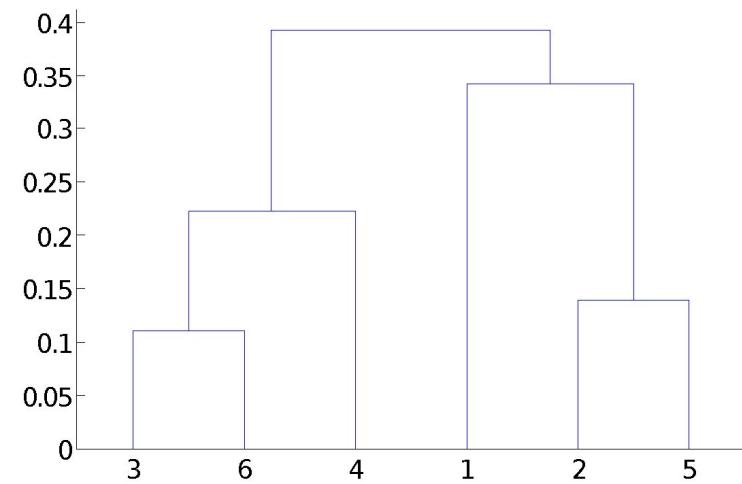
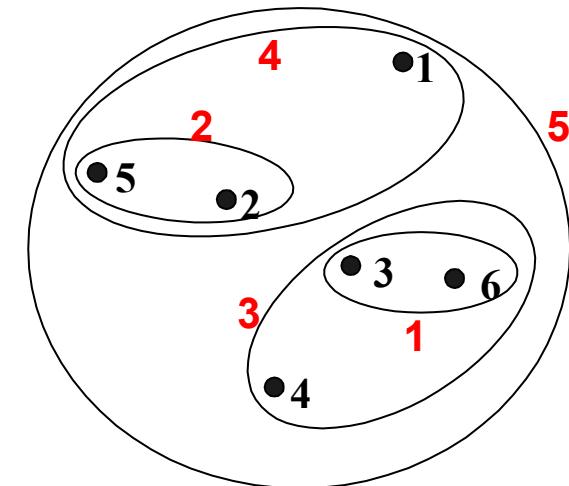
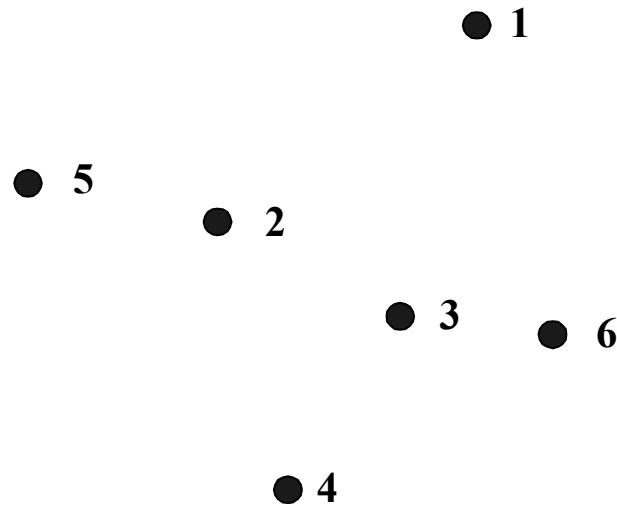
# Agglomerative clustering exercise

- How do clusters change with different linkage methods?
- Single



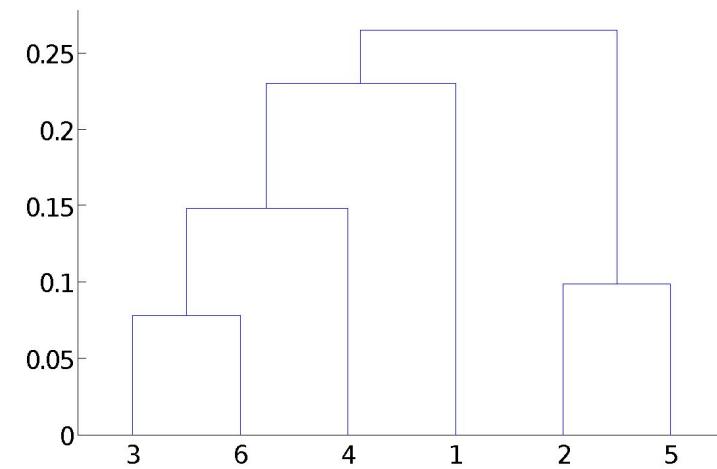
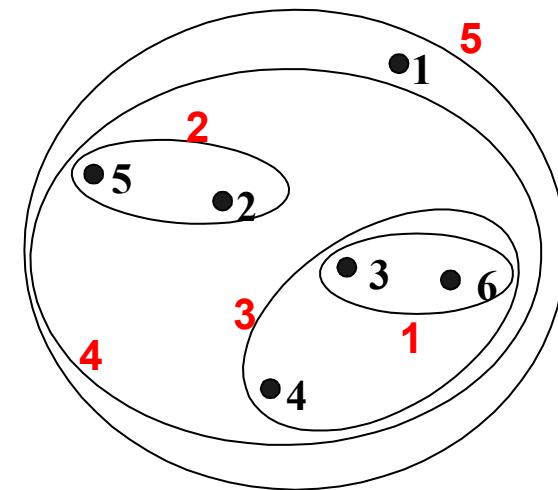
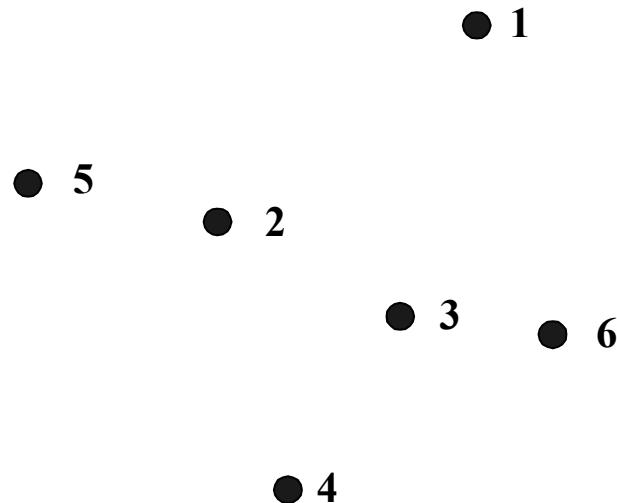
# Agglomerative clustering exercise

- How do clusters change with different linkage methods?
- Complete

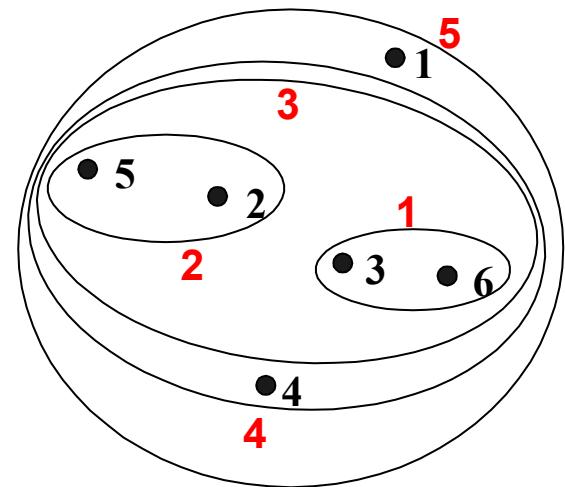


# Agglomerative clustering exercise

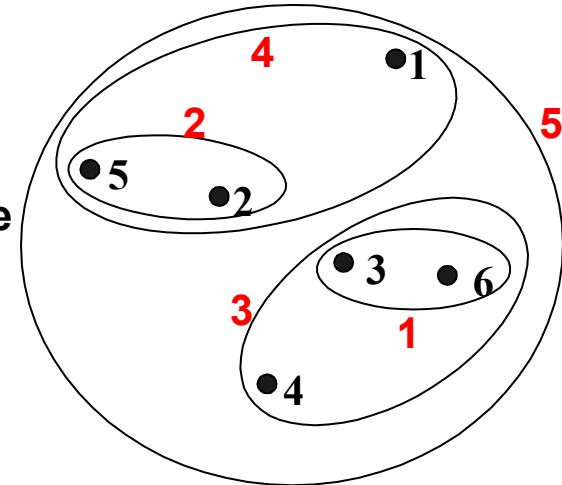
- How do clusters change with different linkage methods?
- Average



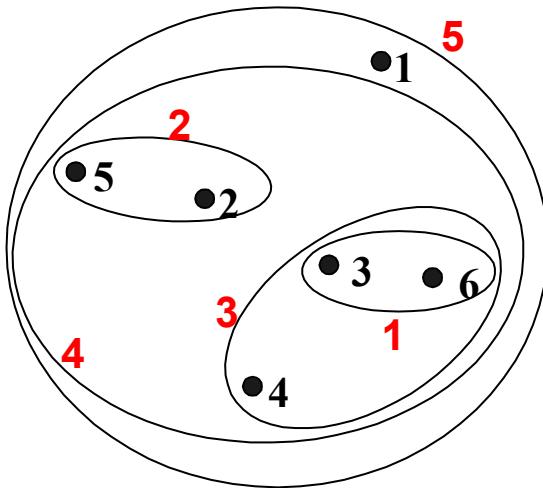
# Linkage Comparison



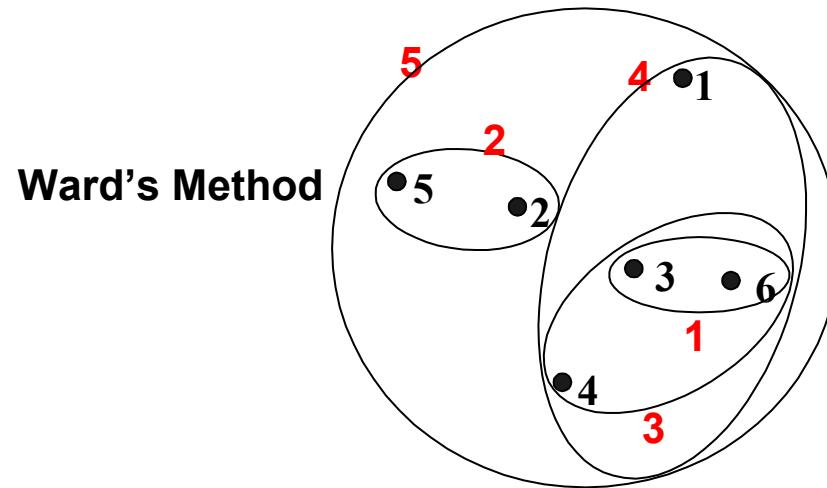
Single



Complete



Average



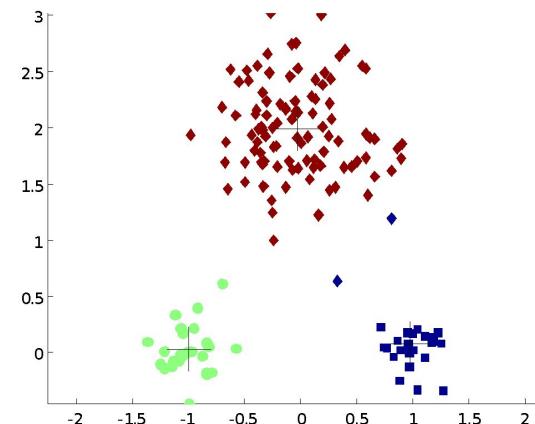
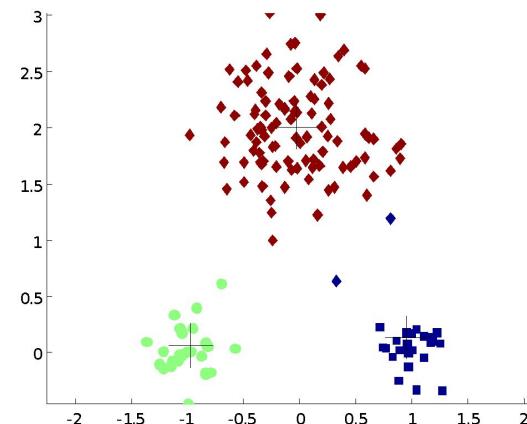
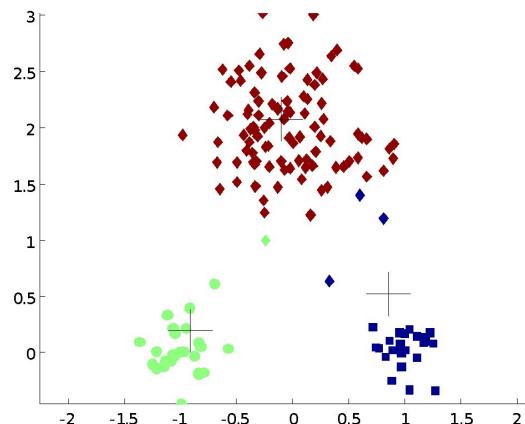
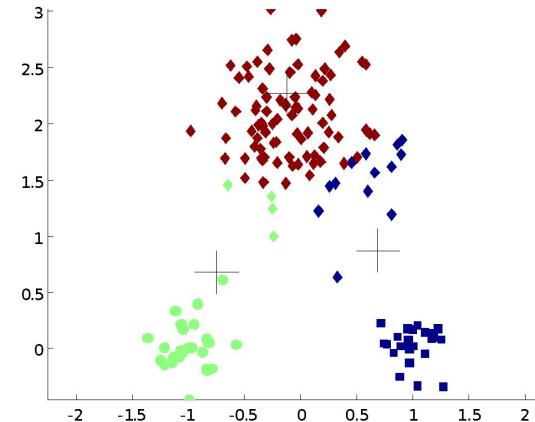
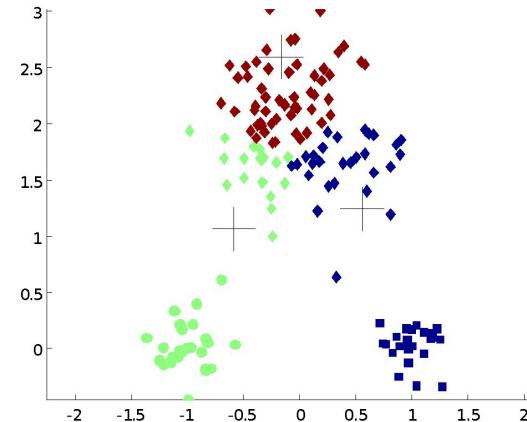
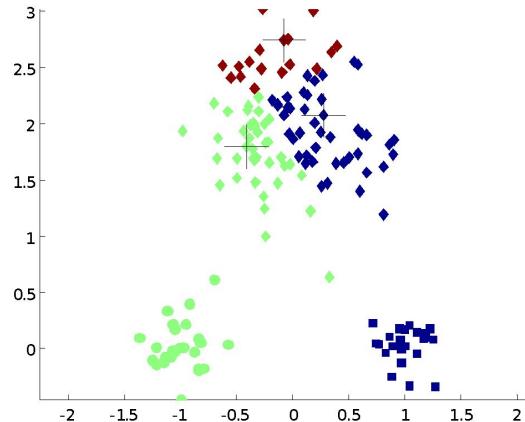
Ward's Method

# K-means clustering

- Partition clustering approach
- Number of clusters ( $K$ ) must be specified
- Each cluster is associated with a centroid
- Each datapoint is assigned to the cluster with the closest centroid

- 
- 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
-

# Example of K-means clustering



# More on K-means clustering

- Initial centroids often chosen randomly
  - Clusters will vary from one run to the next
- Centroid is typically the mean of the points in the cluster
- ‘Closeness’ is measured by similarity metric (e.g., Euclidean distance)
- Convergence usually happens within first few iterations

sự hội tụ thường xảy ra chỉ với vài vòng lặp

# Evaluating K-means clusters

Most common measure is Sum of Squared Error (SSE)

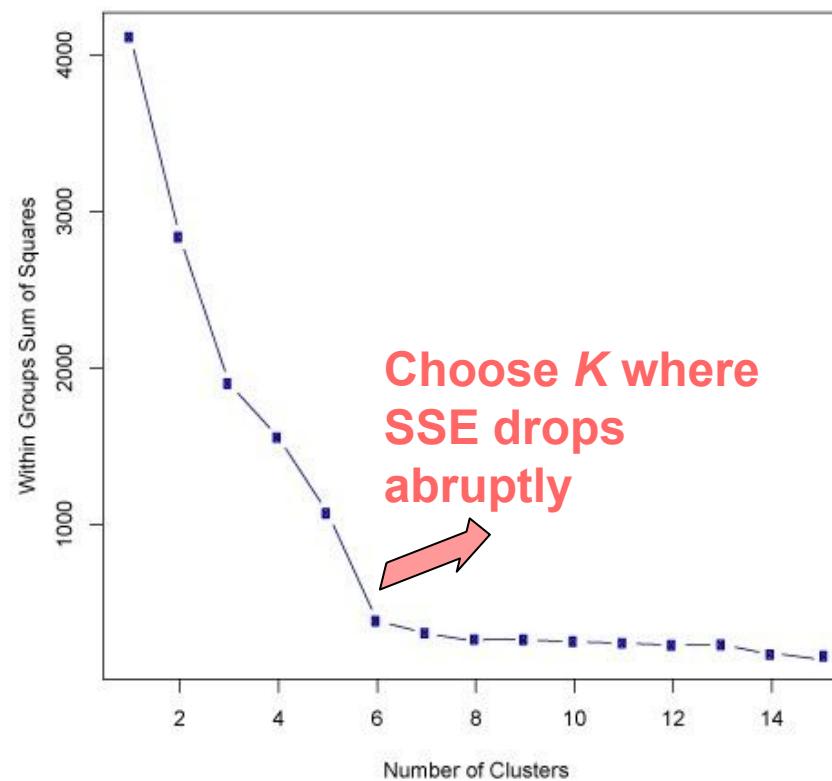
- SSE is the sum of the squared distance between each member of the cluster and the cluster's centroid:

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(m_i, x) \quad m = \text{centroid in cluster } C_i \\ x = \text{a data point in cluster } C_i$$

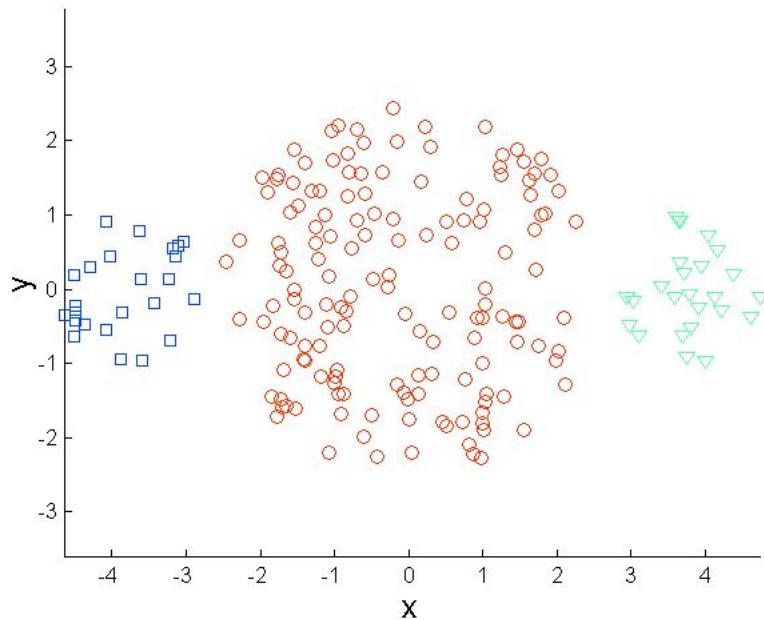
- Given two sets of clusters, we prefer the one with the **smallest error**
- One way to reduce SSE is to increase  $K$   
Although, a good clustering with small  $K$  can have a lower SSE than a poor clustering with high  $K$

# Choosing $K$

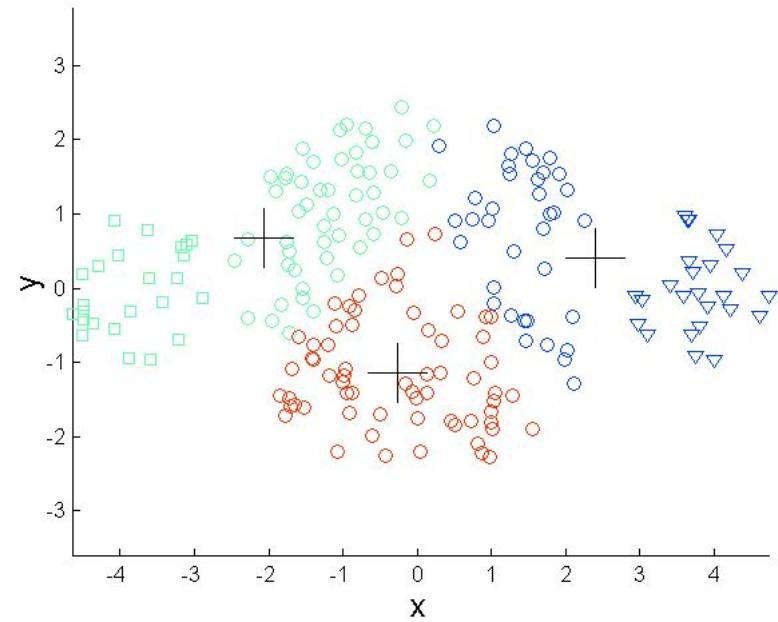
- Visual inspection
- “Elbow method”



# Limitations of K-means: Different sizes

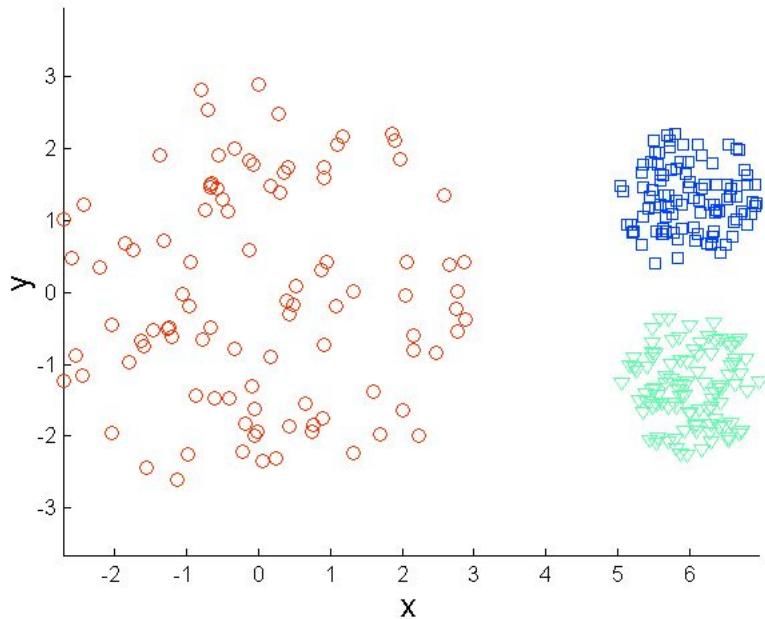


Original Points

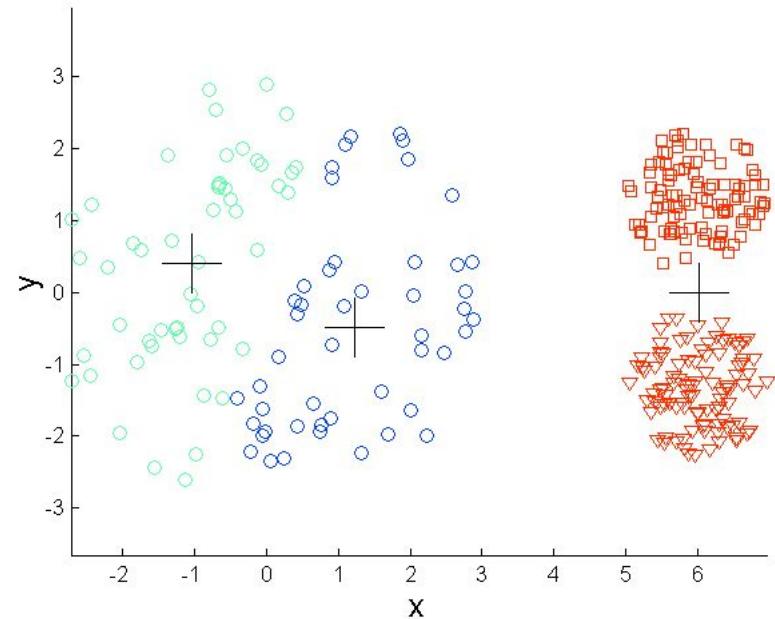


K-means (3  
Clusters)

# Limitations of K-means: Differing density

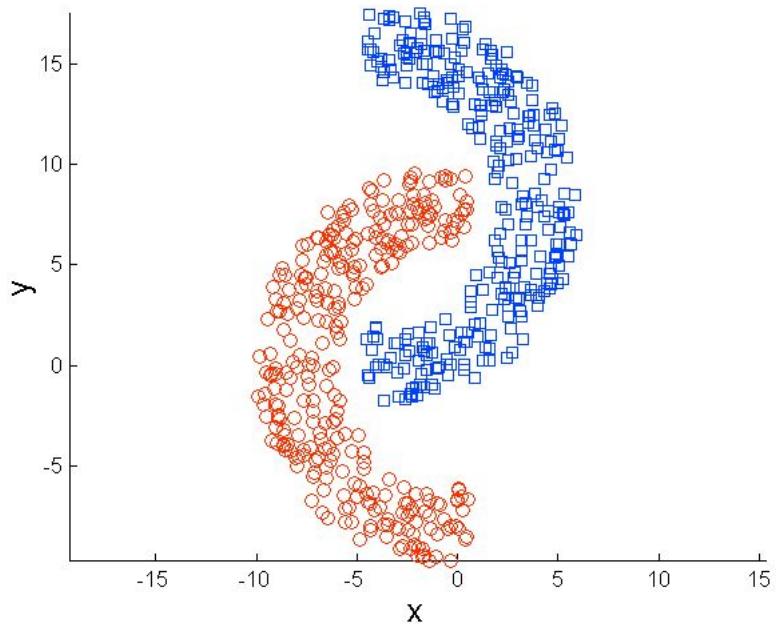


Original Points

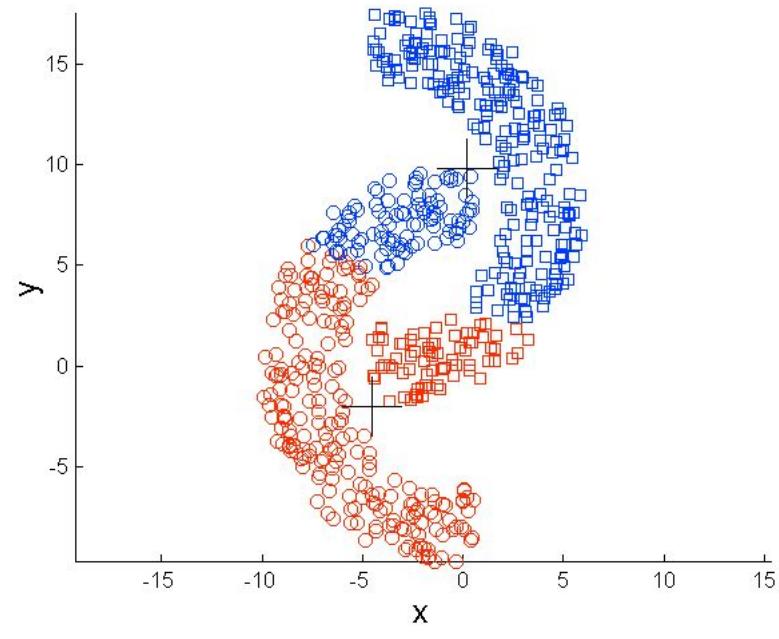


K-means (3  
Clusters)

# Limitations of K-means: Non-globular shapes

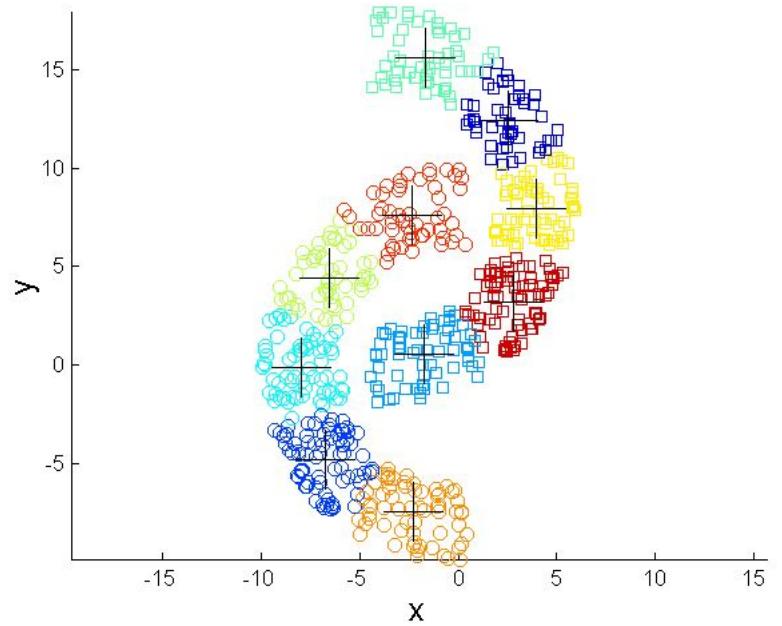
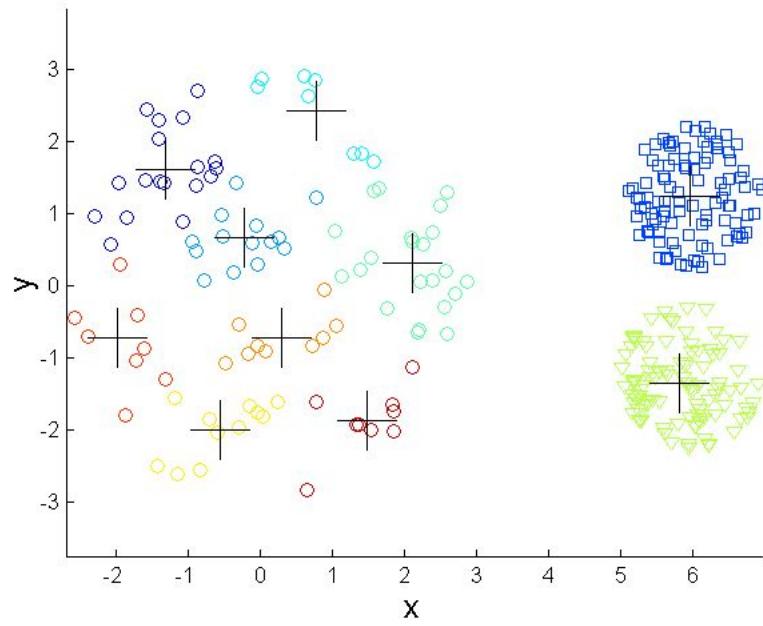


**Original Points**



**K-means (2  
Clusters)**

# Overcoming K-means Limitations



One solution is to use many clusters.  
Find parts of desired clusters, but need to put together.

# Concerns with selecting initial centroids

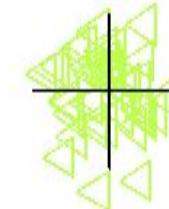
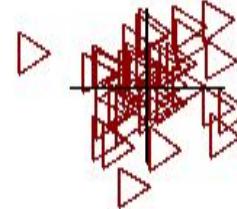
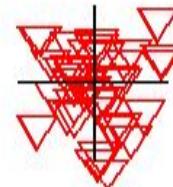
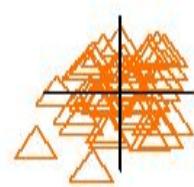
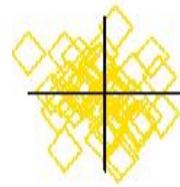
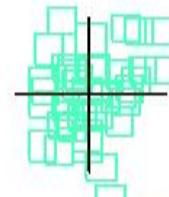
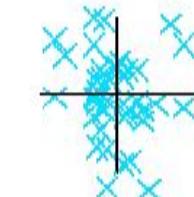
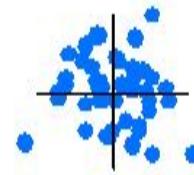
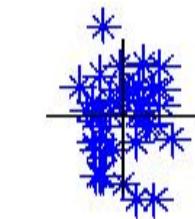
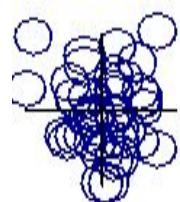
- If there are  $K$  “real” clusters, then the chance of initially selecting one centroid from each cluster is small

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

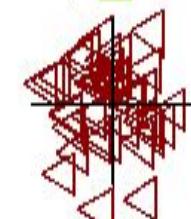
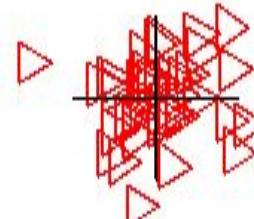
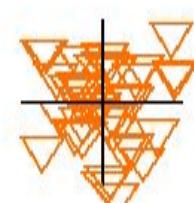
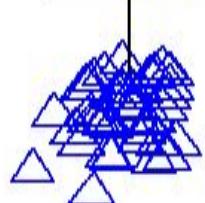
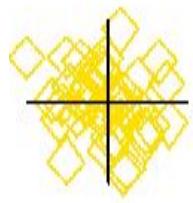
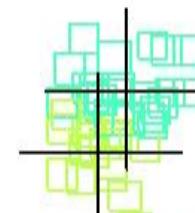
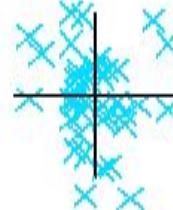
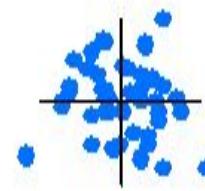
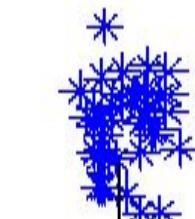
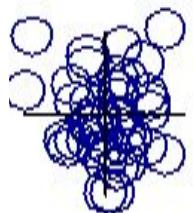
$n$  = size of clusters (assuming relatively similar)

- If  $K=10$ , then  $P = 10!/10^{10} = 0.00036$
- Consider an example of ten clusters....

“Real” clusters:



Clusters obtained with K=10, some “real” clusters without initial centroids:



# Solving initial centroids issues

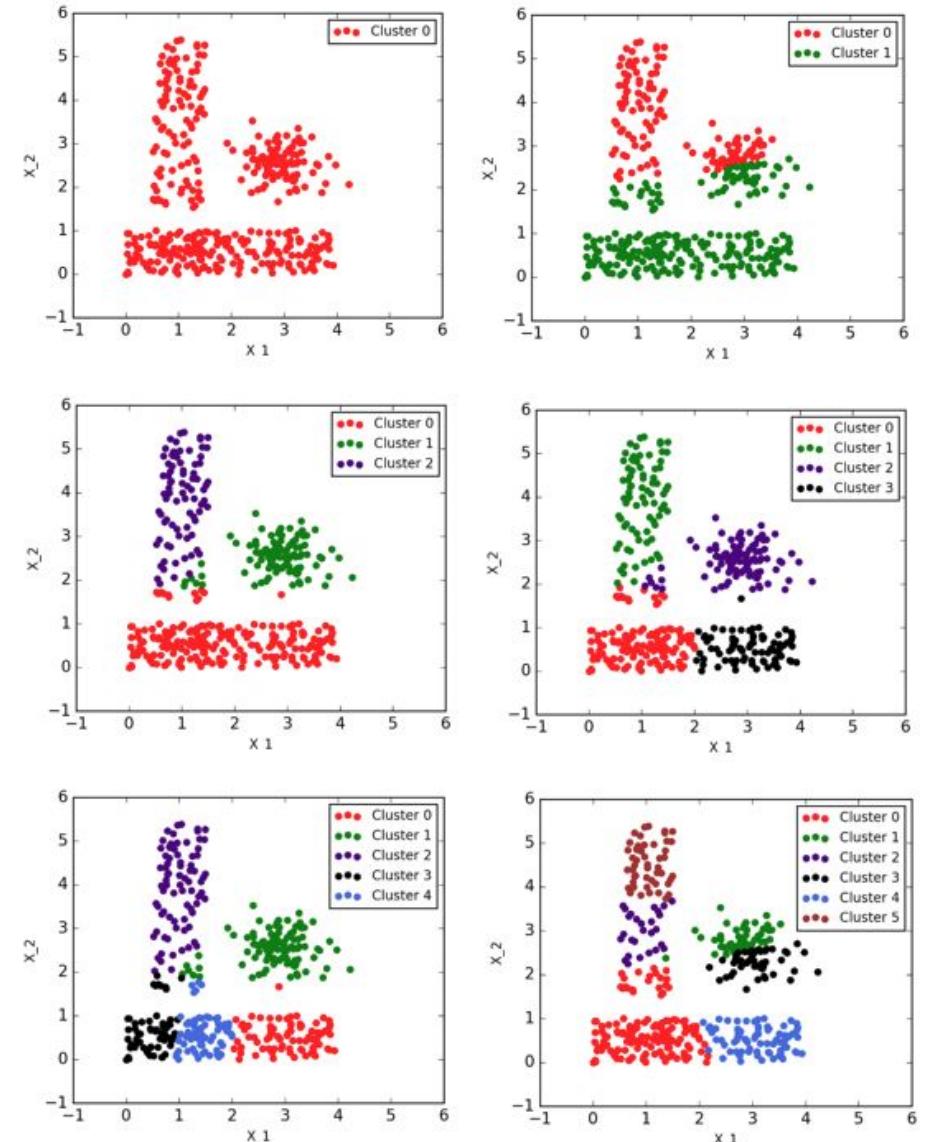
- Multiple runs
- Use hierarchical clustering to determine initial centroids
- Select more than  $K$  initial centroids, then subselect among these (select most widely separated)
- Post-processing
- Generate a larger number of clusters, then perform hierarchical clustering
- Use Bisecting K-means

# Pre- and Post-processing

- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters (may represent outliers)
  - Split ‘loose’ clusters (i.e., clusters w/ high SSE)
  - Merge clusters that are close (w/ low SSE)

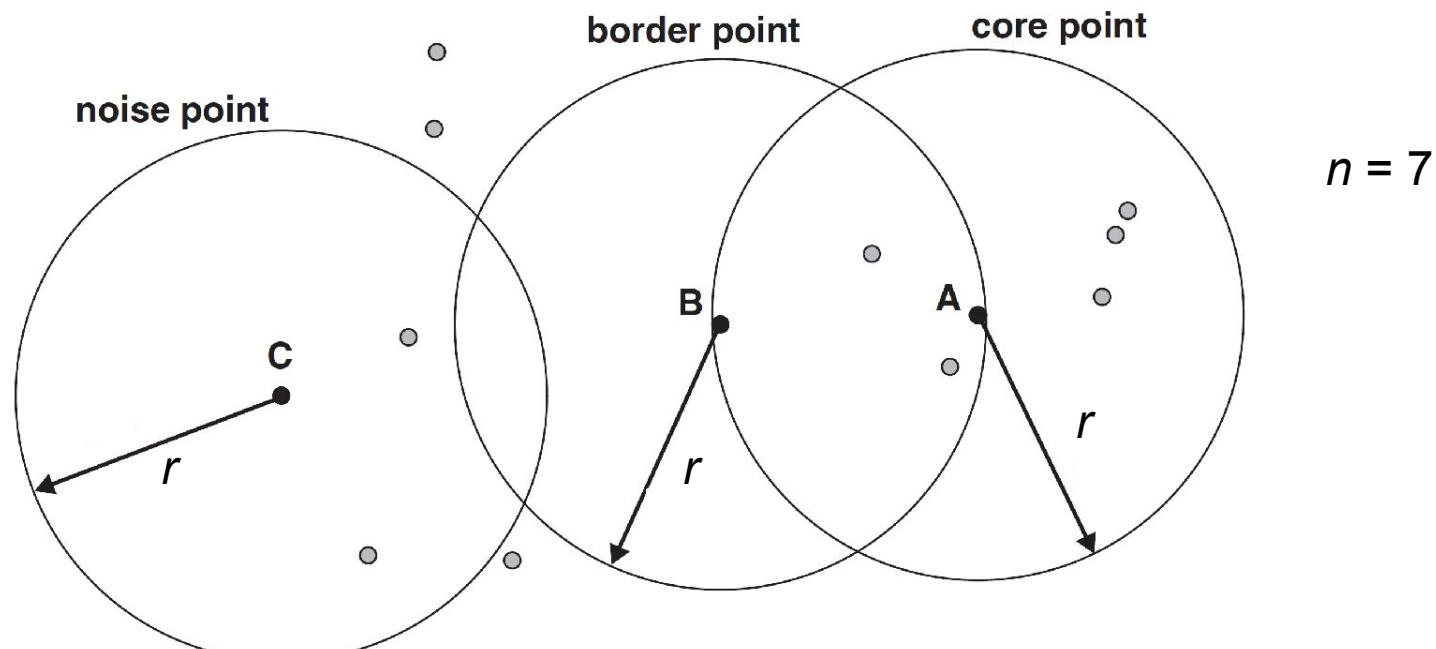
# Bisecting K-means

- Combines K-means and hierachial clustering
- Clusters are iteratively split via regular K-means with  $K=2$
- Stops when desired # of clusters is reached



# Density-based clustering

- Assumes clusters are areas of high density separated by areas of low density
- **Core points** are in areas of a certain density (at least  $n$  points in radius  $r$  from the core point)
- **Border points** aren't core points, but are w/in  $r$  of the core point
- **Noise points** are all other points



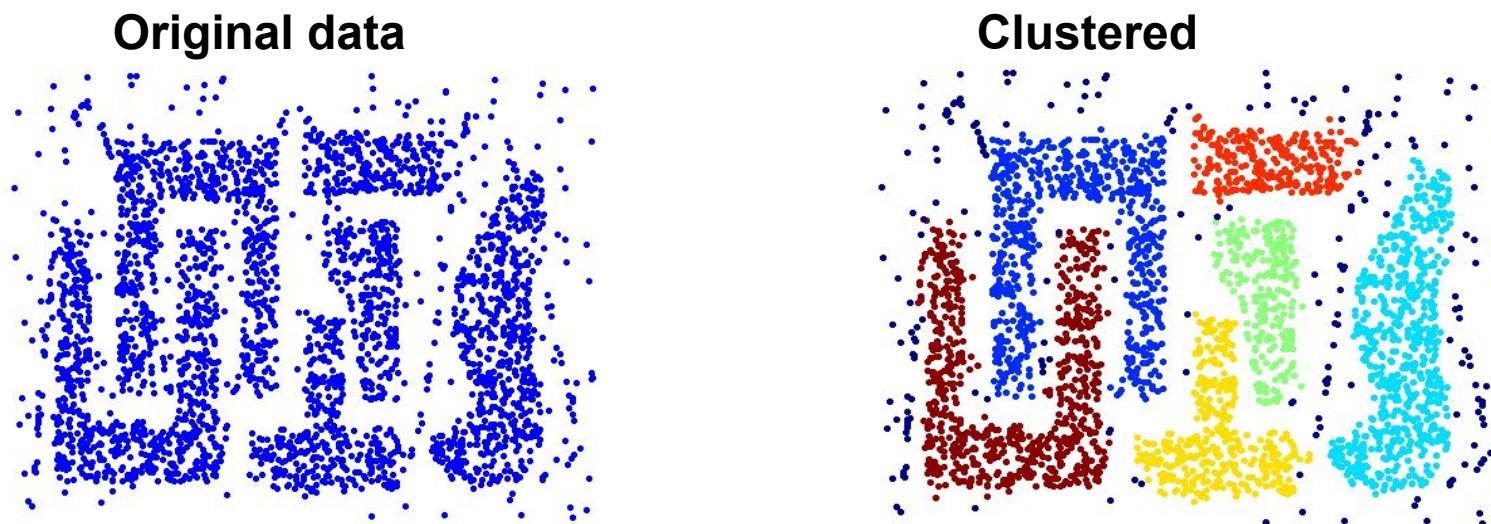
# DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on remaining points

```
current_cluster_label ← 1
for all core points do
    if the core point has no cluster label then
        current_cluster_label ← current_cluster_label + 1
        Label the current core point with cluster label current_cluster_label
    end if
    for all points in the  $Eps$ -neighborhood, except  $i^{th}$  the point itself do
        if the point does not have a cluster label then
            Label the point with cluster label current_cluster_label
        end if
    end for
end for
```

# DBSCAN Advantages & Limitations

- **Advantages:**
- Resistant to noise
- Can handle clusters of different shapes and sizes
- Number of clusters is determined by the algorithm



## Limitations:

- Struggles to identify clusters with varying densities – clustering is often incomplete at points in low density regions are ignored
- Density can be difficult/expensive to compute in high-dimensional datasets

“Clusters are in the eye of the beholder”

*But we might want to evaluate them anyway*

# Outline

- Background
  - Intro
  - Workflow
  - Similarity metrics
- Clustering algorithms
  - Hierarchical
  - K-means
  - Density-based
- **Cluster evaluation**
  - External
  - Internal

# Cluster validation

- 1) Determining the **clustering tendency** of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
- 2) Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
- 3) Comparing the results of two different sets of cluster analyses to determine which is better.
- 4) Determining the ‘correct’ number of clusters.

For 2 and 3, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

# External measures of cluster validity

**External Index:** Extend to which cluster labels match externally supplied class labels

- e.g., gene functional groups, tissue of origin
- *F*-measure provides assessment of cluster **purity and completeness**
  - Purity: fraction of a cluster taken up by predominant class label
  - Completeness: fraction of items in the class grouped in the cluster at hand
- Rand index compares similarity between two clusterings, or known vs predicted labels

# Internal measures of cluster validity

**Internal Index:** Measures goodness of clustering without respect to external info

- How compact are the clusters?
  - SSE
  - Average/maximum pairwise intra-cluster distances
- How well separated are the clusters?
  - Average inter-cluster distance
  - Minimum separation between individual clusters

“The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”

*Algorithms for Clustering Data*, Jain and Dubes

