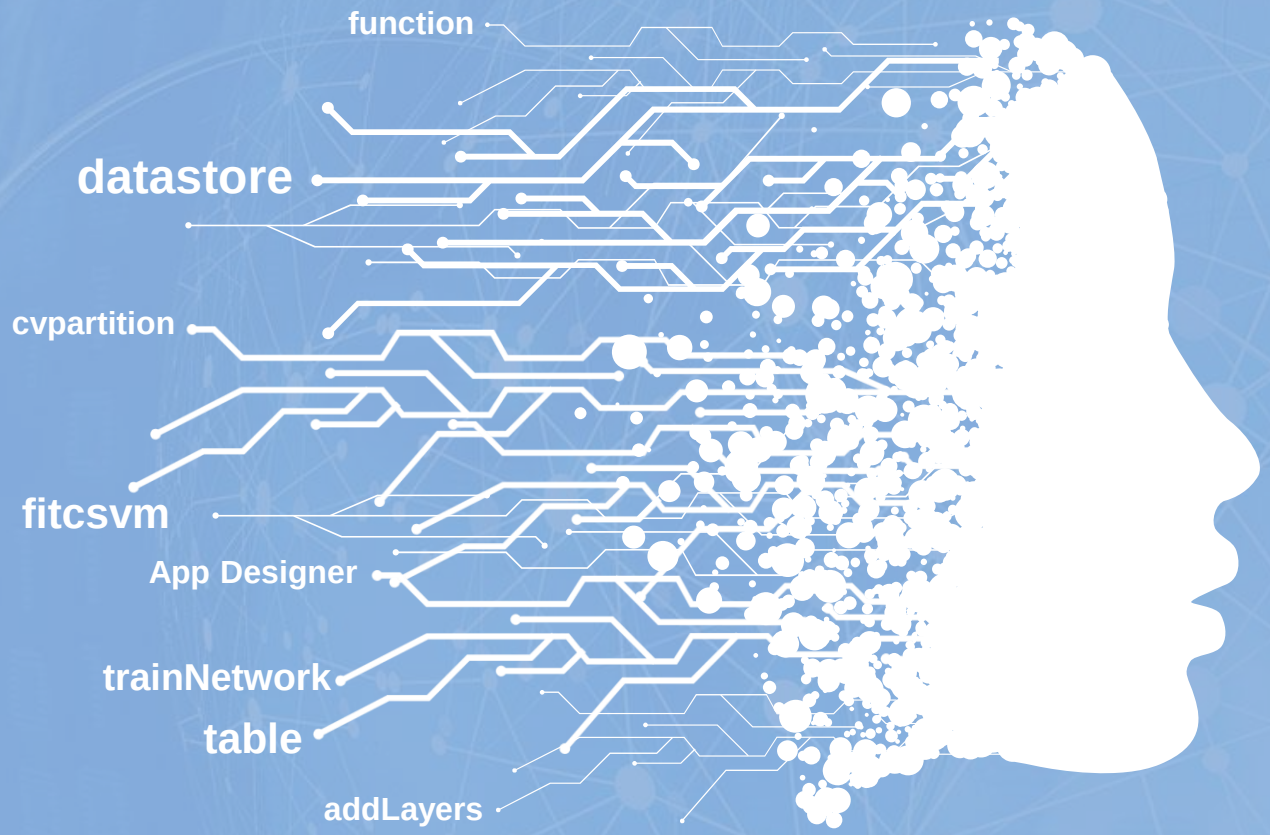
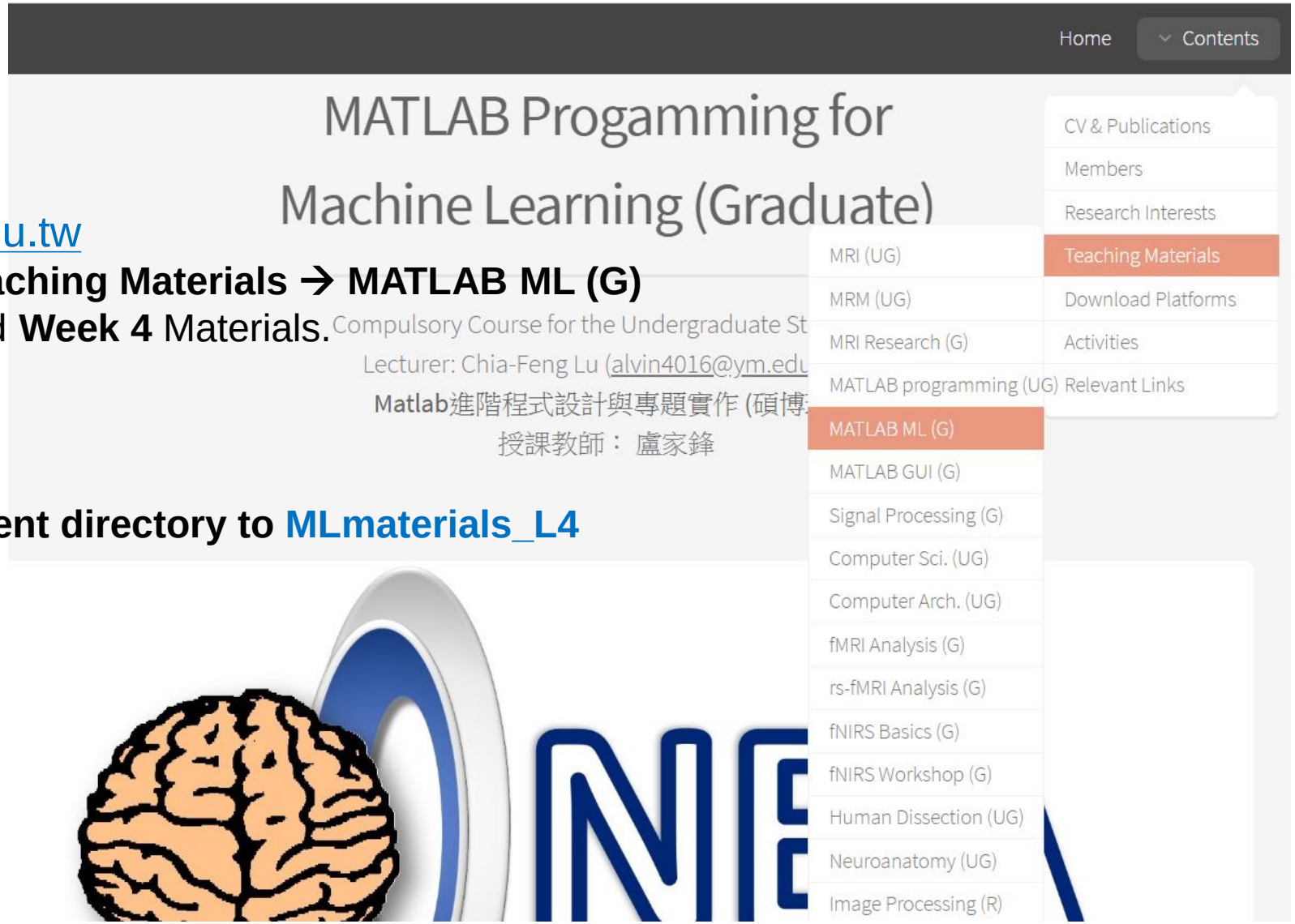


Unsupervised Clustering

MATLAB進階程式語言與實作

盧家鋒 Chia-Feng Lu, Ph.D.
Department of Biomedical Imaging and Radiological Sciences, NYCU
alvin4016@nycu.edu.tw

Teaching Materials



cflu.lab.nycu.edu.tw

Contents → Teaching Materials → MATLAB ML (G)

Please download **Week 4** Materials.

Please set current directory to MLmaterials_L4

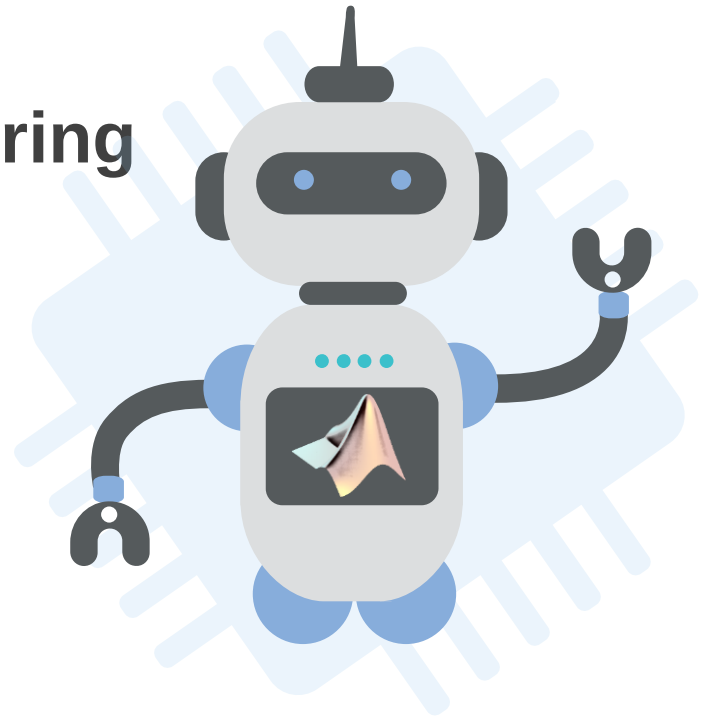
Contents in this Week

01 **Unsupervised Learning: Hard Clustering**

K-means, hierarchical clustering

02 **Unsupervised Learning: Soft Clustering**

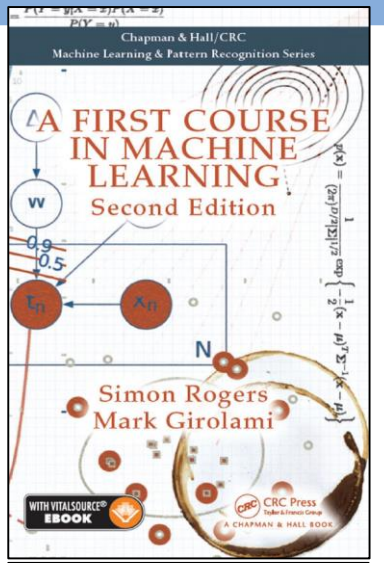
Mixture models



References

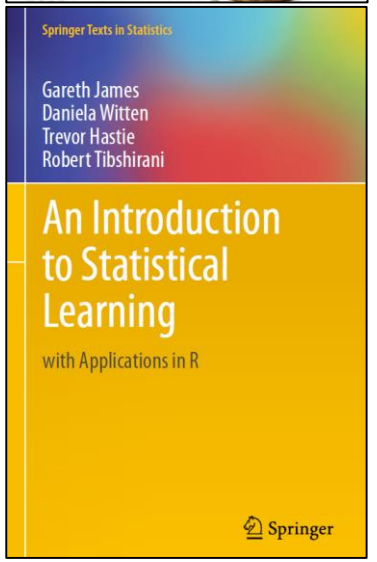
[Textbook 1]

- **A First Course in Machine Learning, 2nd edition, 2017**
  Simon Rogers, Mark Girolami
- **Online resources:** https://github.com/sdrogers/fcmlcode
- **K-Means, Mixture Models**



[Textbook 3]

- **An Introduction to Statistical Learning, 2nd edition, 2013**
  Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
- **Online resources:** https://github.com/rghan/ISLR
- **K-Means, Hierarchical Clustering**



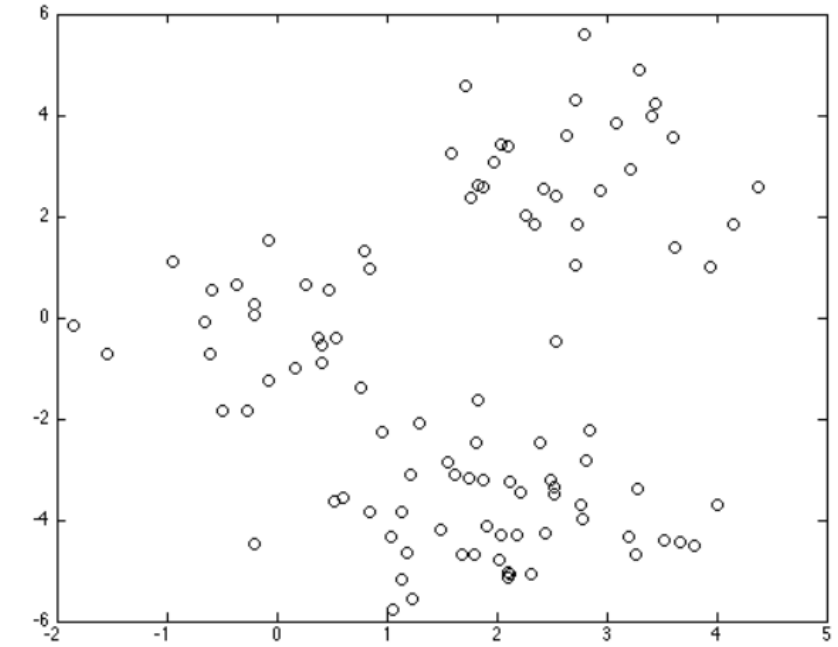
Unsupervised Learning: Hard Clustering

K-means, hierarchical clustering

Unsupervised Learning: Clustering

- The goal is to discover interesting things about the measurements on $x_1, x_2, \ldots, x_n$.
- Can we discover **subgroups** among the variables or among the observations?

- We may have a reason to believe that there is some heterogeneity among the *n* observation samples.
- **Clustering** looks to find homogeneous subgroups among the observations.



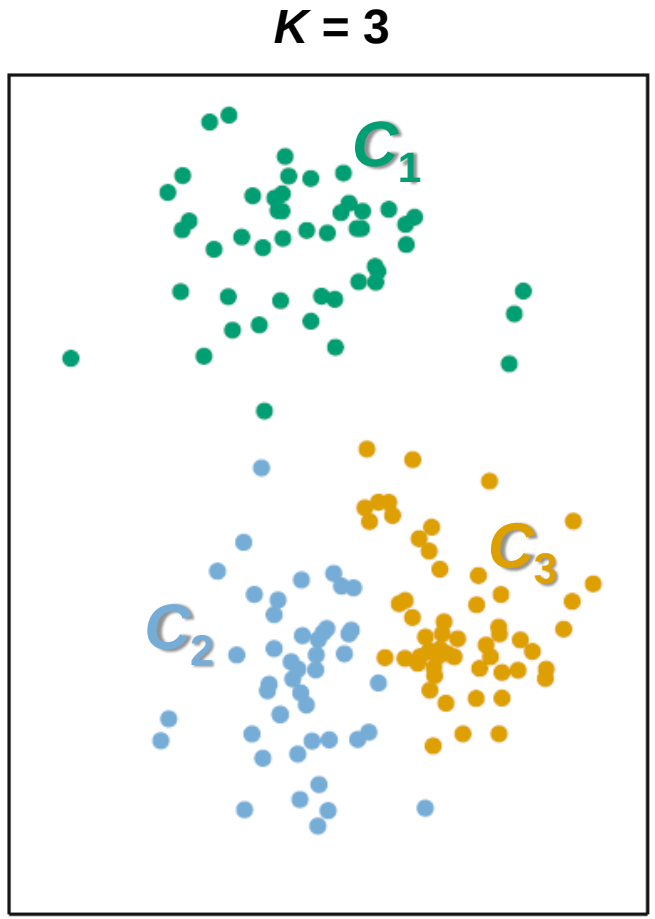
K-Means Clustering

- We must first specify the desired number of clusters ***K***.

- Let $C_1, \ldots, C_K$ denote sets containing the indices of the observations in each cluster.

  1. $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \ldots, n\}$
     Each observation belongs to at least one of the clusters.
  2. $C_k \cap C_{k'} = \emptyset, for\ all\ k \neq k'$
     The clusters are non-overlapping.
     No observation belongs to more than one cluster.

< Definition of hard clustering >



K-Means Clustering

- A *good* clustering should have the *within-cluster variation,* $\boldsymbol{W(C_k)}$, to be as small as possible.
- Hence, we want to minimize $\boldsymbol{W(C_k)}$

$$\boldsymbol{arg \min_{C_1,\ldots,C_k} \left\{ \sum_{k=1}^{K} W(C_k) \right\}}$$

- The most common estimate of $\boldsymbol{W(C_k)}$ is *squared Euclidean distance*.

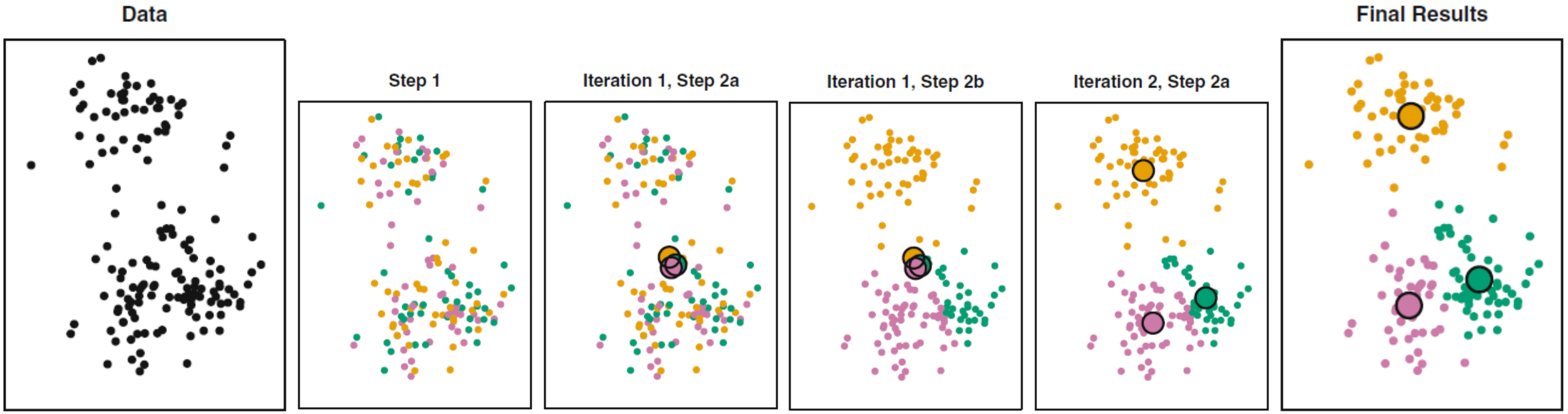
$$\boldsymbol{W(C_k) = \frac{1}{N_k} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2}$$

$\boldsymbol{X_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]}$

The *i*th sample $\boldsymbol{X_i}$ includes **p features**

Algorithm of K-Means Clustering

- Assign cluster randomly
- Compute the cluster centroid
- Re-assign the cluster based on the closest centroid



Algorithm of K-Means Clustering

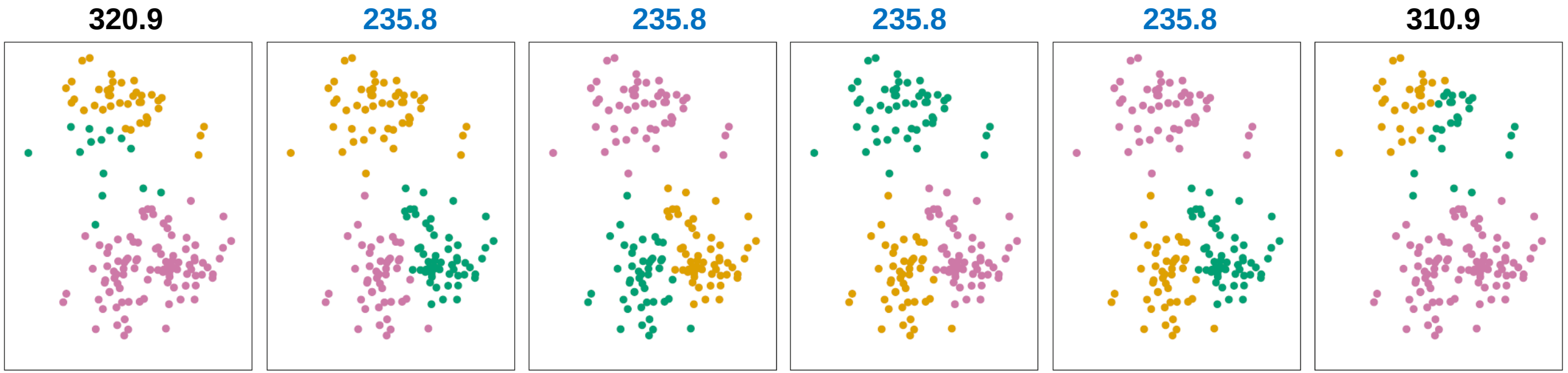
1. Randomly assign a number, from 1 to *K*, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:

   (a) For each of the *K* clusters, compute the cluster ***centroid***. The *k*th cluster centroid is the vector of the *p* feature means for the observations in the *k*th cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where ***closest*** is defined using Euclidean distance).

Avoiding Local Optimum

- Because the clustering results depend on the initial (random) cluster assignment, a local rather than global optimum may be obtained.



- Run the algorithm multiple times from different random initial configurations ➔ select the best solution with smallest objective.

Exercise – K-Means Clustering

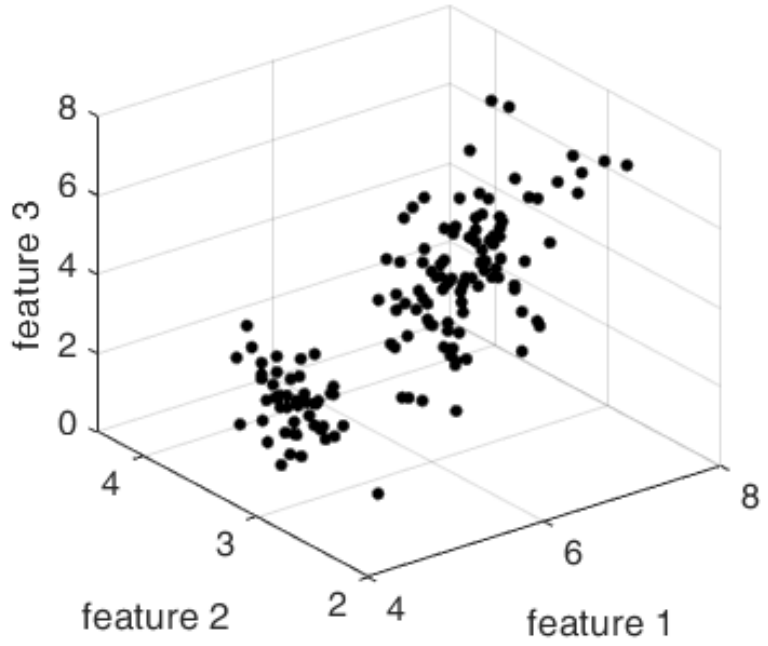
- Partition data into ***K*** mutually exclusive clusters.
- Fisher's Iris dataset (load fisheriris)
- 50 samples from each of three species of Iris
- Four features were measured from each sample: the length and the width of the sepals and petals (in cm)

Iris setosa

Iris versicolor

Iris virginica

150 observations of iris



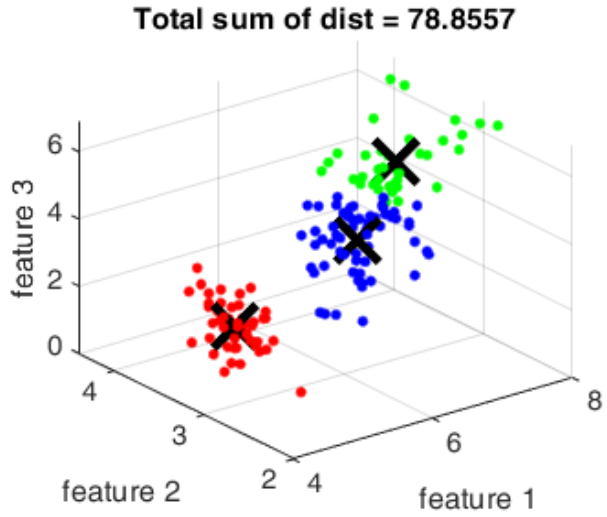
MLmaterials_L4\Ex_Kmeans.m

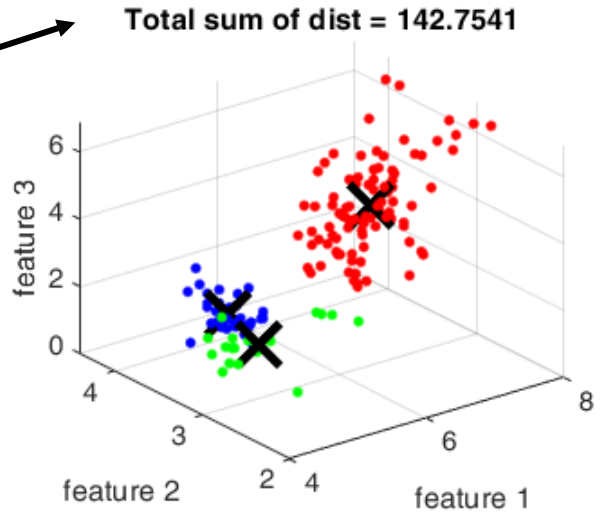
Exercise – K-Means Clustering

- Perform K-means clustering with **20 replicates** and **parallel computing**

```
opts = statset('UseParallel',1);
[ind,C,sumd] = kmeans(meas,3,'MaxIter',10000,…
                      'Replicates',20,'Options',opts);
```

- Replicate 10, 3 iterations, total sum of distances = **78.8514**.
- Replicate 9, 4 iterations, total sum of distances = 78.8514.
- Replicate 14, 10 iterations, total sum of distances = **142.754**.
- Replicate 12, 7 iterations, total sum of distances = 78.8557.
- Replicate 11, 2 iterations, total sum of distances = 78.8557.
- Best total sum of distances = 78.8514

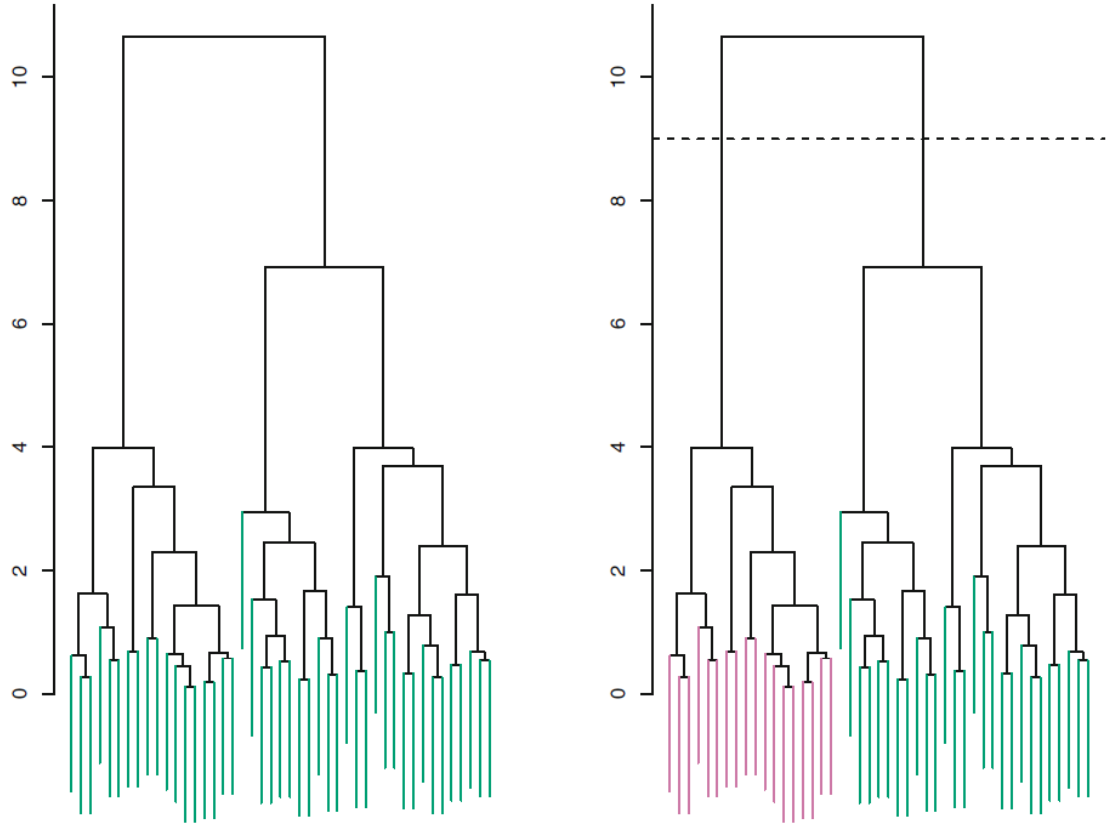




MLmaterials_L4\Ex_Kmeans.m

Hierarchical Clustering

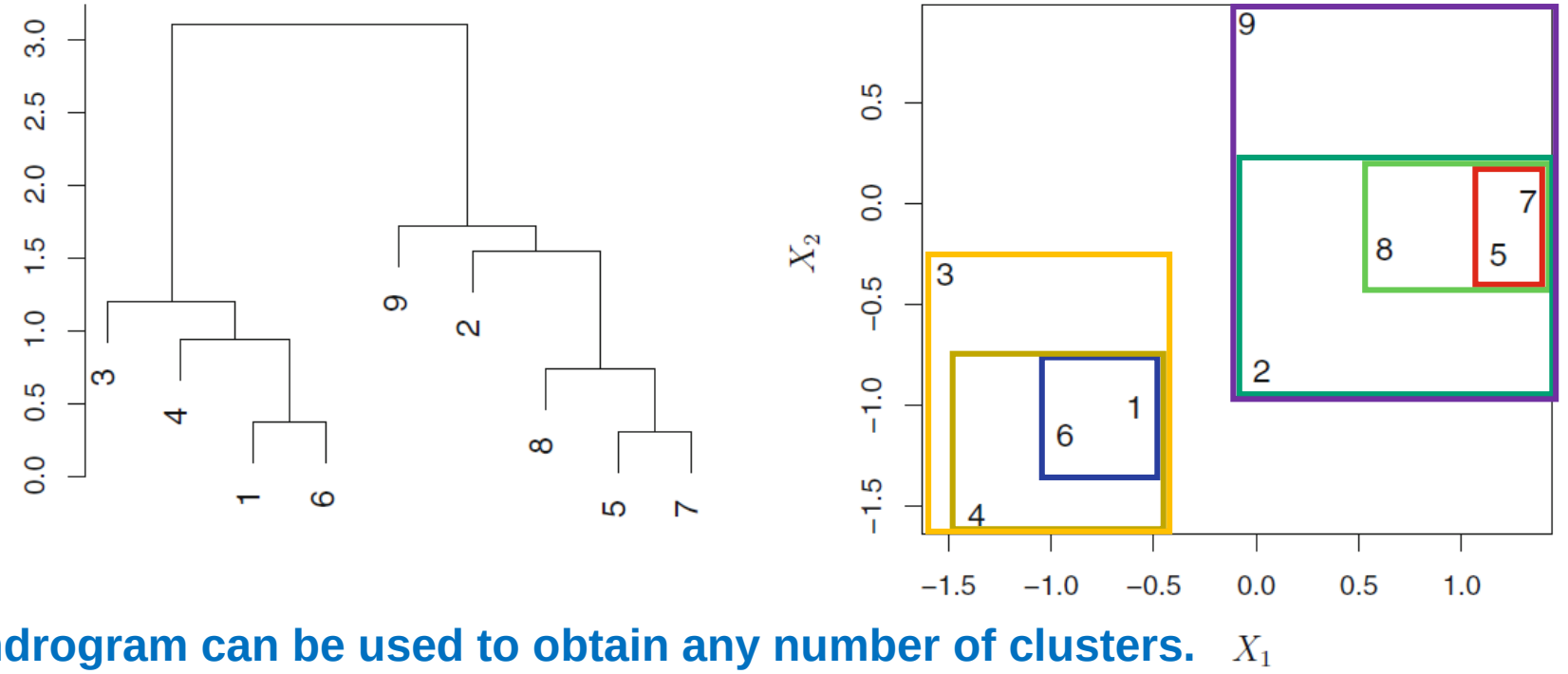
- An alternative approach which does not require that we commit to a particular choice of *K*.
- An added advantage over *K*-means clustering in that it results in an attractive tree-based representation of the observations, called a ***dendrogram***.



45 observations

Interpretation of Dendrogram

- The earlier (lower in the tree) fusions occur, the more similar the groups of observations are to each other.
- the height of the cut to the dendrogram serves controls the number of clusters obtained.



One single dendrogram can be used to obtain any number of clusters.

Algorithm of Hierarchical Clustering

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities.
2. For $k = n, n-1, n-2, \ldots, 2$:

   (a) Examine all pairwise inter-cluster dissimilarities among the *k* clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters.

   The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

   (b) Compute the new pairwise inter-cluster dissimilarities among the *k* − 1 remaining clusters (linkage, dissimilarities between clusters).

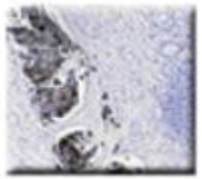
Dissimilarities – Distance metric

Value	Description
'euclidean'	Euclidean distance (default).
'squaredeuclidean'	Squared Euclidean distance. (This option is provided for efficiency only. It does not satisfy the triangle inequality.)
'seuclidean'	Standardized Euclidean distance. Each coordinate difference between observations is scaled by dividing by the corresponding element of the standard deviation, S = nanstd(X). Use DistParameter to specify another value for S.
'mahalanobis'	Mahalanobis distance using the sample covariance of X, C = nancov(X). Use DistParameter to specify another value for C, where the matrix C is symmetric and positive definite.
'cityblock'	City block distance.
'minkowski'	Minkowski distance. The default exponent is 2. Use DistParameter to specify a different exponent P, where P is a positive scalar value of the exponent.
'chebychev'	Chebychev distance (maximum coordinate difference).
'cosine'	One minus the cosine of the included angle between points (treated as vectors).
'correlation'	One minus the sample correlation between points (treated as sequences of values).
'hamming'	Hamming distance, which is the percentage of coordinates that differ.
'jaccard'	One minus the Jaccard coefficient, which is the percentage of nonzero coordinates that differ.
'spearman'	One minus the sample Spearman's rank correlation between observations (treated as sequences of values).
@distfun	Custom distance function handle. A distance function has the form

Linkage Methods

Method	Description
'average'	Unweighted average distance
'centroid'	Centroid distance, appropriate for Euclidean distances only
'complete'	Farthest distance
'median'	Weighted center of mass distance, appropriate for Euclidean distances only
'single'	Shortest distance
'ward'	Inner squared distance (minimum variance algorithm), appropriate for Euclidean distances only
'weighted'	Weighted average distance

Demo Dataset – NCI60



NCI60 Cancer Microarray Project

The web supplement to D.T. Ross *et al.* (2000)
Nature Genetics, 2000 March, 24(3): 227-234

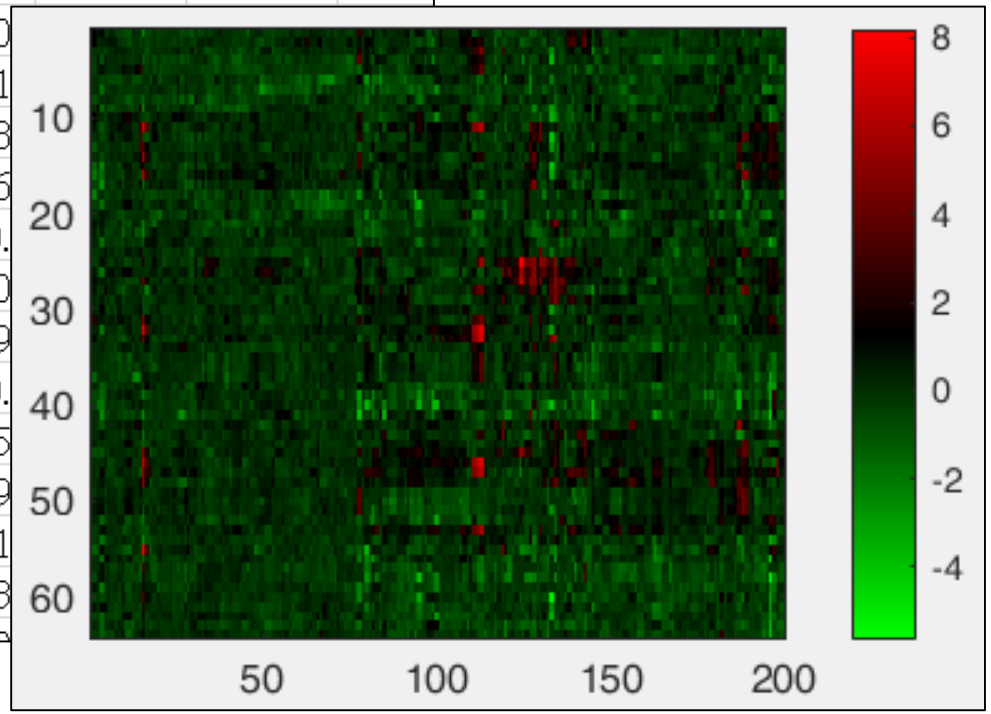
- Cancer Microarray Project
  - http://genome-www.stanford.edu/nci60/
- NCI60 is a dataset of gene expression profiles of **60 National Cancer Institute (NCI) cell lines.**
  - derived from patients with leukaemia, melanoma, lung, colon, central nervous system, ovarian, renal, breast and prostate cancers.
- Ross DT, Scherf U, Eisen MB, Perou CM, Rees C, Spellman P, Iyer V, Jeffrey SS, Van de Rijn M, Waltham M, Pergamenschikov A. **Systematic variation in gene expression patterns in human cancer cell lines**. *Nature genetics*. 2000 Mar;24(3):227-35.

Demo Dataset – NCI60

- MLmaterials_L4\NCI60data.csv

64 x 6830 matrix

	A	B	C	D	E	F	G	H	I	J	K
1		1	2	3	4	5	6	7	8	9	
2	CNS	0.3	1.18	0.55	1.14	-0.265	-0.07	0.35	-0.315	-0.45	-0.65
3	CNS	0.679961	1.289961	0.169961	0.379961	0.464961	0.579961	0.699961	0.724961	-0.04004	-0.28
4	CNS	0.94	-0.04	-0.17	-0.04	-0.605	0	0.0			
5	RENAL	0.28	-0.31	0.68	-0.81	0.625	-1.39E-17	0.1			
6	BREAST	0.485	-0.465	0.395	0.905	0.2	-0.005	0.08			
7	CNS	0.31	-0.03	-0.1	-0.46	-0.205	-0.54	-0.6			
8	CNS	-0.83	0	0.13	-1.63	0.075	-0.36	0.			
9	BREAST	-0.19	-0.87	-0.45	0.08	0.005	0.35	-0.0			
10	NSCLC	0.46	0	1.15	-1.4	-0.005	-0.7	-0.9			
11	NSCLC	0.76	1.49	0.28	0.1	-0.525	0.36	0.			
12	RENAL	0.27	0.63	-0.36	-1.04	0.015	-0.04	0.5			
13	RENAL	-0.45	-0.06	0.15	-0.61	-0.395	0.15	0.9			
14	RENAL	-0.03	-1.12	-0.05	0	-0.285	-0.25	0.1			
15	RENAL	0.71	0	0.16	-0.77	0.045	-0.16	0.3			



Exercise – Hierarchical Clustering

- **Find the similarity or dissimilarity between every pair of objects in the data set.**

  >> pdist

- **Group the objects into a binary, hierarchical cluster tree.**

  >> linkage

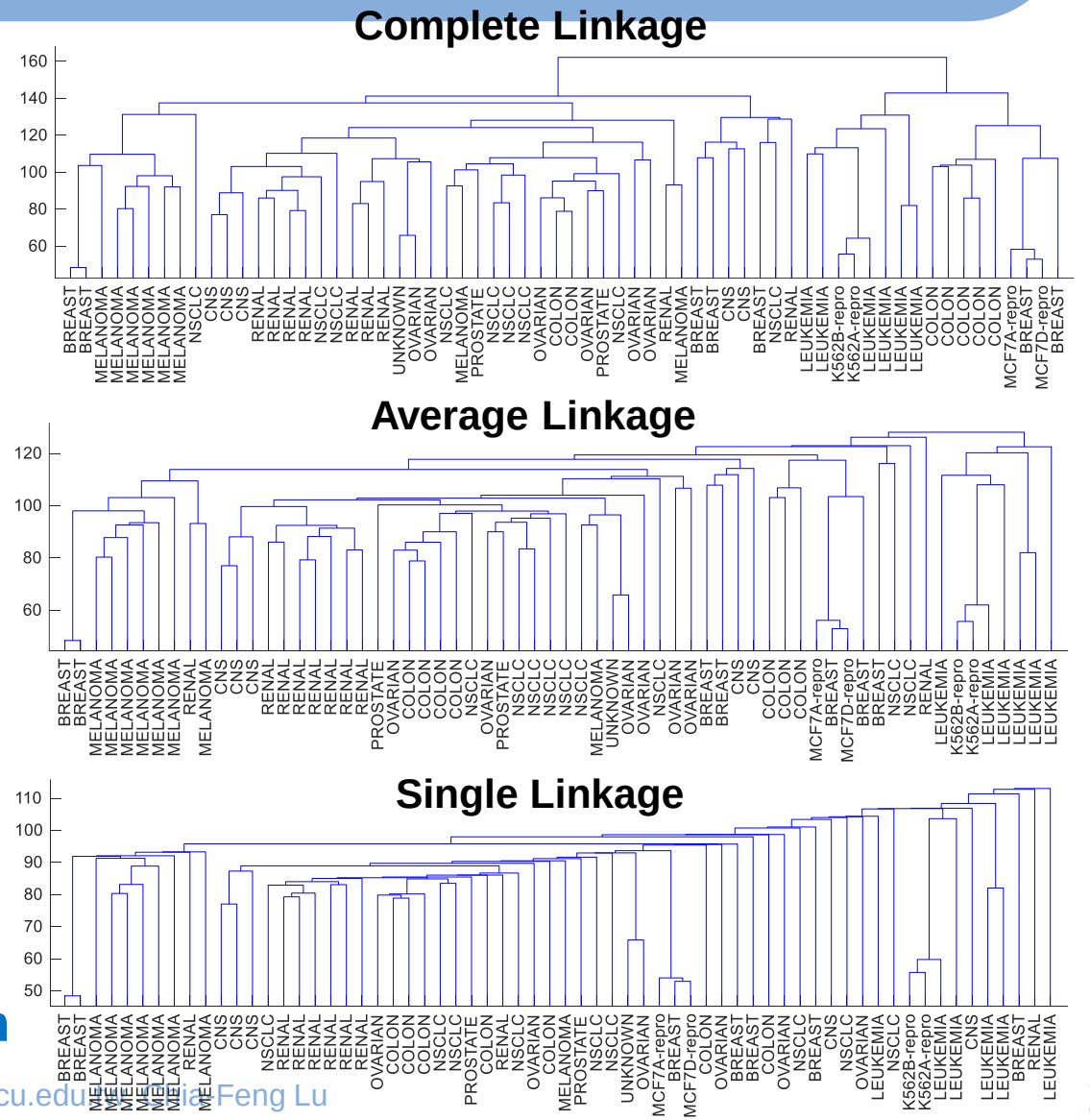
- **Determine where to cut the hierarchical tree into clusters.**

  >> cluster

MLmaterials_L4\Ex_HierarchicalClustering.m

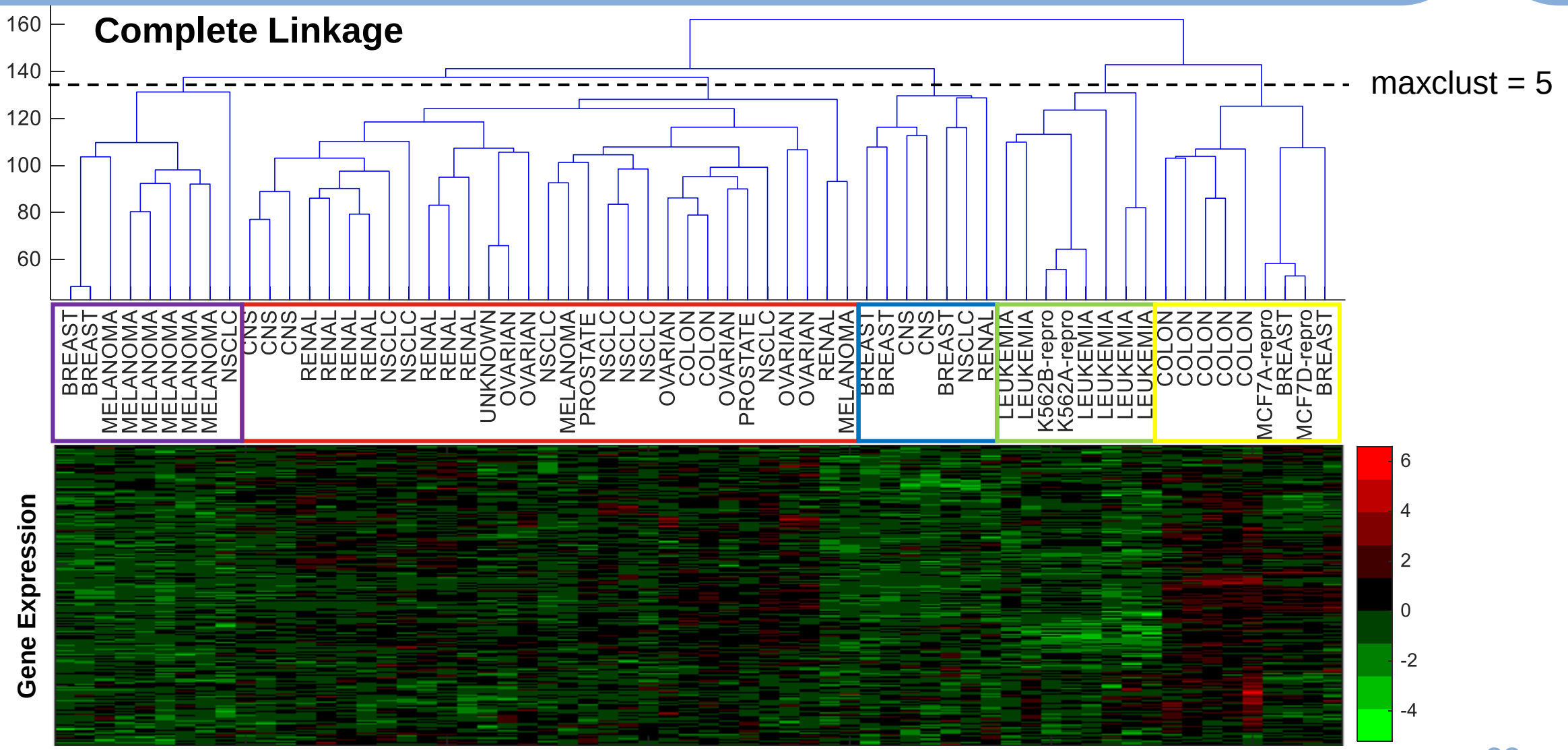
Exercise – Hierarchical Clustering

- **Average** and **complete** linkage tend to yield more balanced clusters.
- **Single** linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.



MLmaterials_L4\Ex_HierarchicalClustering.m

Exercise – Hierarchical Clustering

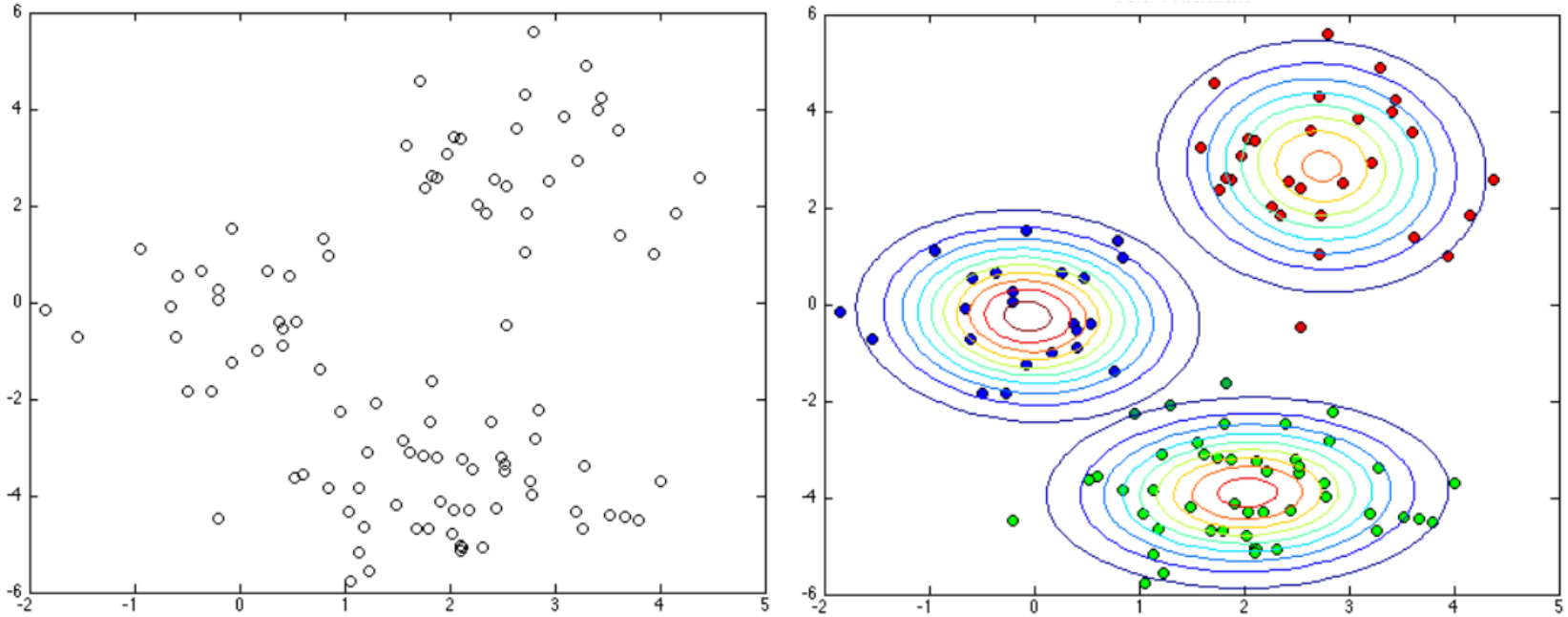


Unsupervised Learning: Soft Clustering

Mixture models

Mixture Models

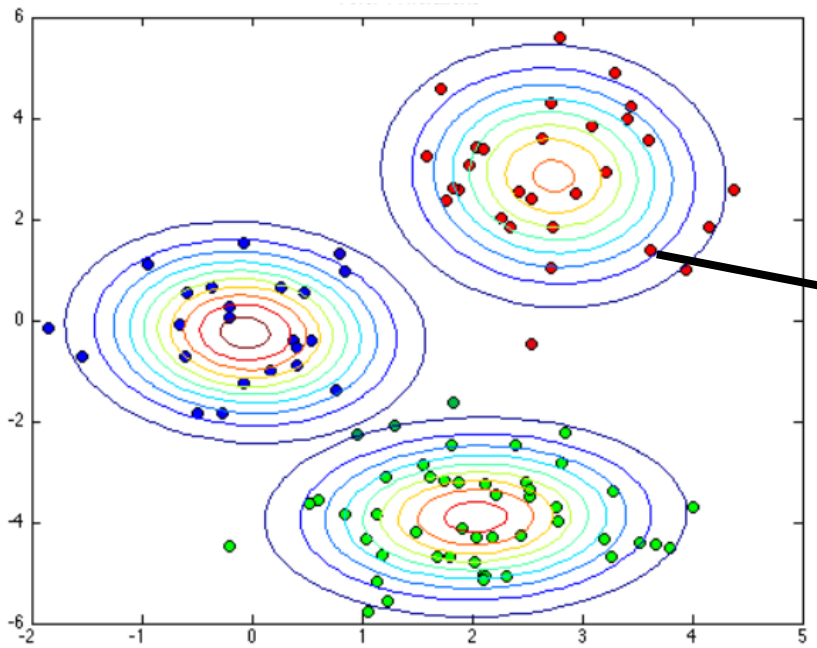
- Instead of exclusive clusters, statistical mixtures represent each cluster as a probability density.
- We can model clusters with a wide variety of shapes in almost any type of data.



Gaussian Mixture Models

- The density function for $\boldsymbol{x_n}$, given that it is from the $\boldsymbol{k}$th component ($\boldsymbol{z_{nk}}$=1), is a Gaussian with **mean $\boldsymbol{\mu_k}$** and **covariance $\boldsymbol{\Sigma_k}$**.

$$p(x_n|z_{nk}=1,\mu_k,\Sigma_k)=N(\mu_k,\Sigma_k)$$



$\mu_3=[0,0],\Sigma_3=\begin{bmatrix}1.5 & 0\\ 0 & 2\end{bmatrix}$

$\mu_1=[3,3],\Sigma_1=\begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix}$

$\pi_1=0.7,\pi_2=0.2,\pi_3=0.1$

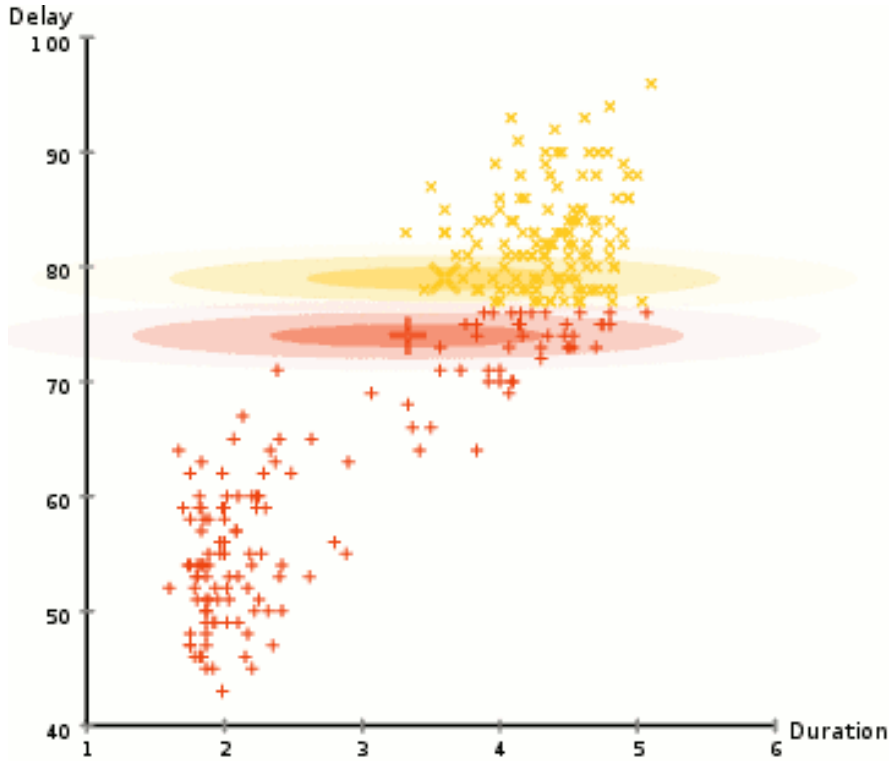
Probability for each cluster

$\mu_2=[2,-4],\Sigma_2=\begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}$

Expectation-Maximization (EM) Algorithm

- **An expectation (E) step**
  - creates a function for the expectation of the **log-likelihood** evaluated using the current estimate for the parameters, and
- **A maximization (M) step**
  - computes parameters maximizing the expected log-likelihood found on the *E* step.

$$L = \sum_{n=1}^{N} log \sum_{k=1}^{K} \pi_k \, p(x_n | \mu_k, \Sigma_k)$$



https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm

Exercise – Gaussian Mixture Models

- Create simulated data from a mixture of two bivariate Gaussian distributions.

$$\mu_1 = [1, 2], \Sigma_1 = \begin{bmatrix} 3 & 0.2 \\ 0.2 & 2 \end{bmatrix}$$

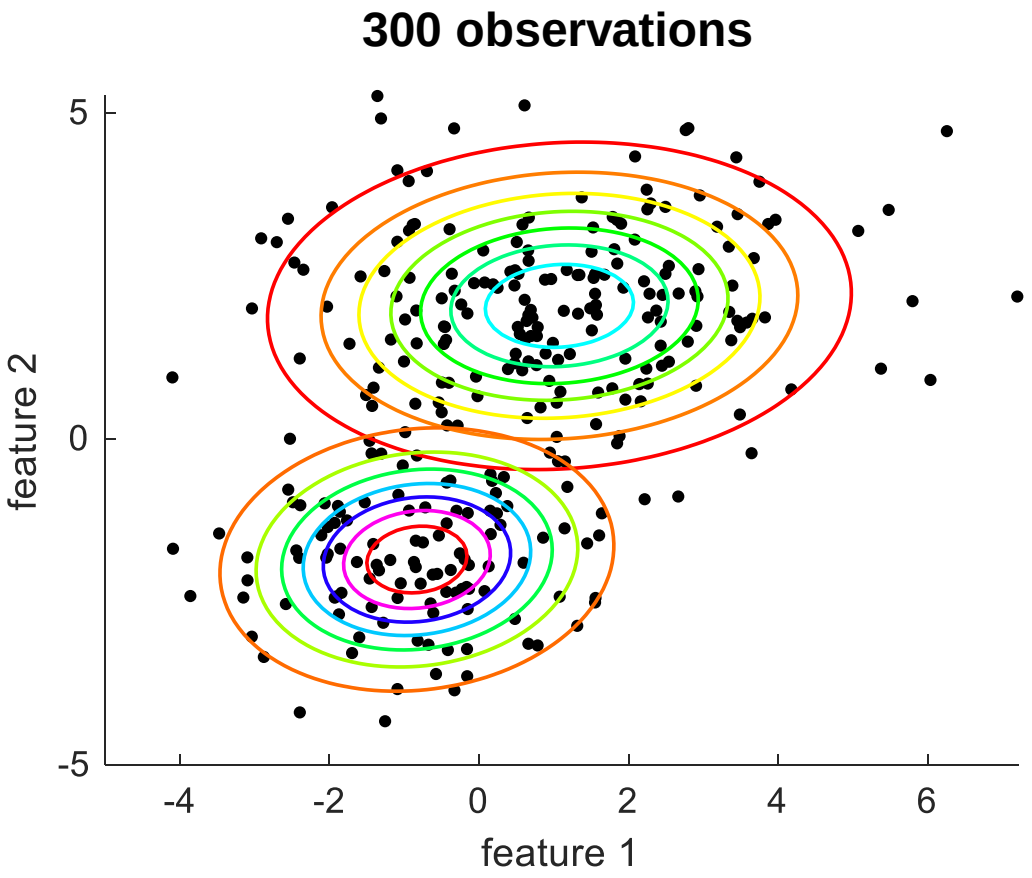
$$\mu_2 = [-1, -2], \Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

- **Fit a two-component Gaussian mixture model**

  >> **gm = fitgmdist(X,K);**

$$\mu_1' = [1.08, 2.04], \Sigma_1' = \begin{bmatrix} 3.66 & 0.18 \\ 0.18 & 1.51 \end{bmatrix}$$

$$\mu_2' = [-0.83, -1.85], \Sigma_2' = \begin{bmatrix} 1.67 & 0.13 \\ 0.13 & 0.98 \end{bmatrix}$$

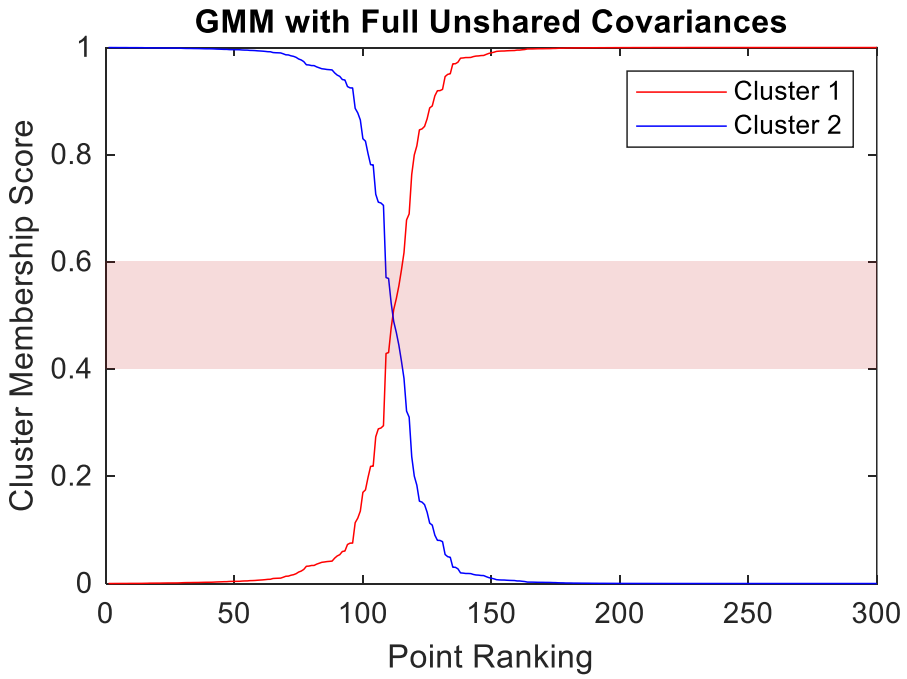


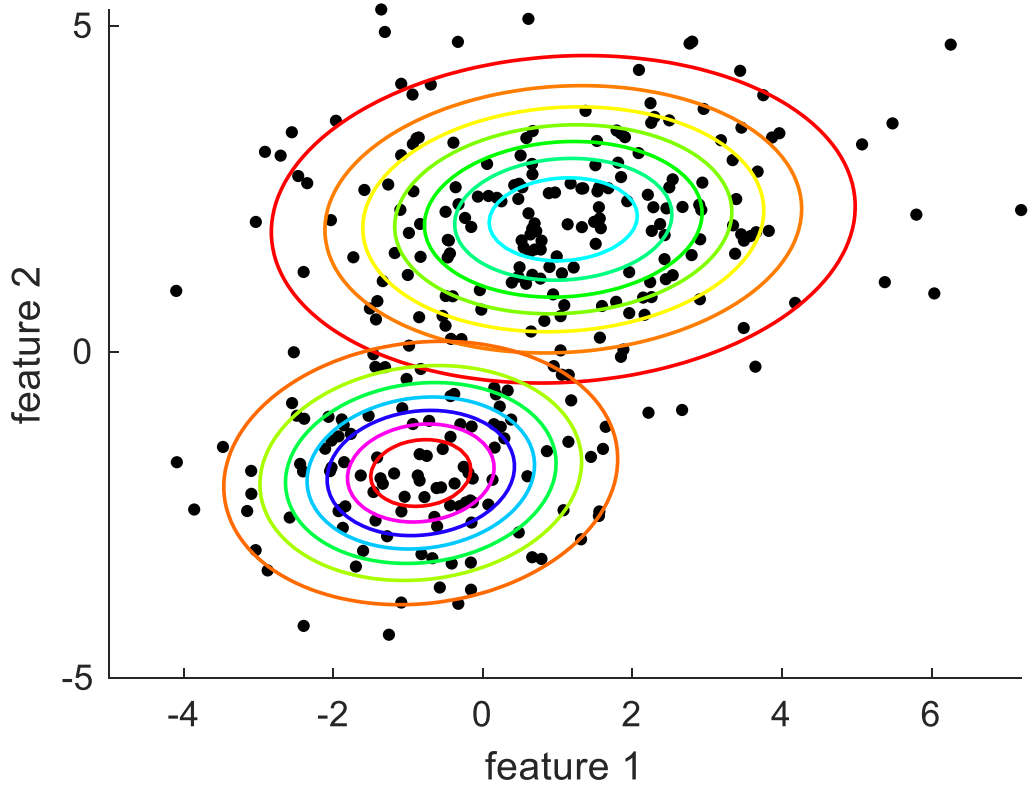
MLmaterials_L4\Ex_GaussianMixModel.m

Exercise – Posterior Probabilities

- Estimate component-member posterior probabilities for all data points

  >> **P = posterior(gm,X);**





MLmaterials_L4\Ex_GaussianMixModel.m

Exercise – Soft Clustering

- Assign clusters by maximum posterior probability.

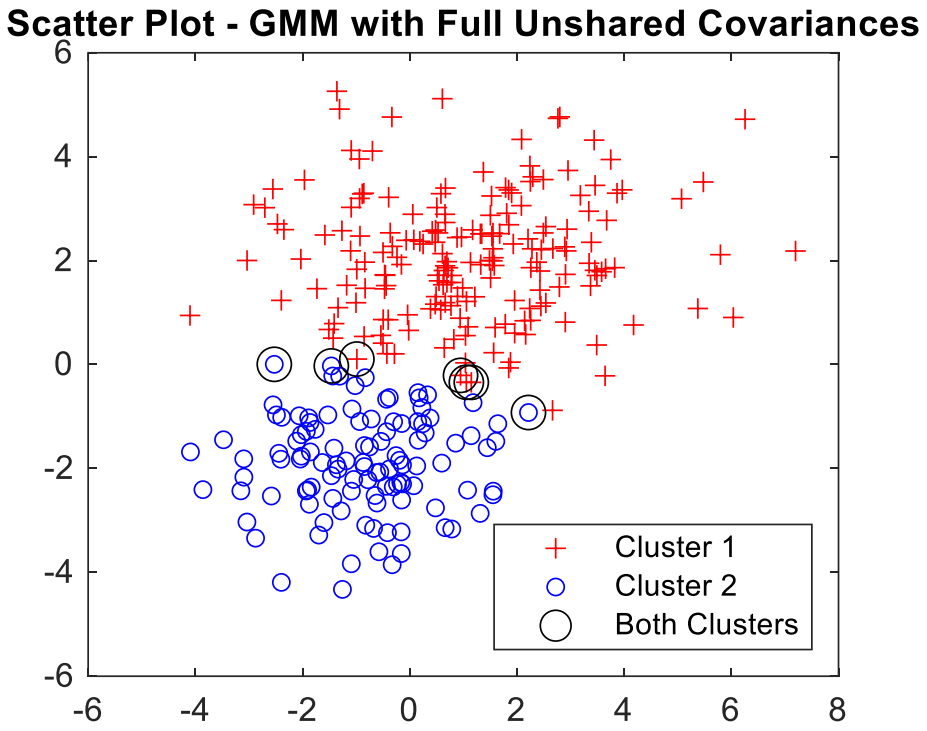
  ind = cluster(gm,X);

- Identify points that could be in either cluster.

  threshold = [0.4 0.6];

  indBoth = find(P(:,1)>=threshold(1)…

  & P(:,1)<=threshold(2));



MLmaterials_L4\Ex_GaussianMixModel.m

THE END

Contact:
盧家鋒 alvin4016@nycu.edu.tw