

Cluster Analysis

Lars Arvastson and Line Clemmensen

March 13, 2018

Today's Lecture

- ▶ Recap
- ▶ What is cluster analysis
- ▶ Similarity and dissimilarity measures
- ▶ K-means clustering
- ▶ Hierarchical clustering
- ▶ Gaussian mixture
- ▶ Validation and model selection

Recap

PCA - Principal Component Analysis

- ▶ What for?
- ▶ How?



Recap

PCR - Principal Component Regression

- ▶ What for?
- ▶ How?



Recap

PLS - Partial Least Squares Regression

- ▶ What for?
- ▶ How?



Recap

CCA - Canonical Correlation Analysis

- ▶ What for?
- ▶ How?



Cluster Analysis

What is cluster analysis?

What are the different types of cluster analysis?

How do I choose the right type of cluster analysis?

What are the steps in cluster analysis?

What are the limitations of cluster analysis?

What are the applications of cluster analysis?

What are the challenges of cluster analysis?

What are the future directions of cluster analysis?

What are the best practices for cluster analysis?

What are the common mistakes in cluster analysis?

What are the key takeaways from cluster analysis?

Cluster Analysis

Unsupervised classification

- ▶ Separating or clustering observations
- ▶ Intuitive but vague definition

Given an underlying set of points, partition them into a collection of **clusters** so that points in the same cluster are close together, while points in different clusters are far apart.

Purpose

- ▶ Seeing structure in data
 - ▶ Gaining understanding
- ▶ Dimensionality reduction
- ▶ Outlier detection

Similarity and dissimilarity measure

In what sense are points close in one cluster and far from points in another cluster?

Similarity takes a large value when points are close.

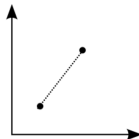
Dissimilarity takes a large value when points are far apart. This reflects the **distance** between observations.

Any monotone-decreasing function can convert similarities to dissimilarities.

Both similarity and dissimilarity measures can be subjective. For example comparing the taste of three ice creams.

Euclidean distance

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2}$$

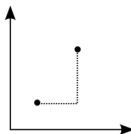


Useful for quantitative variables

Ordinal variables can be transformed to a quantitative scale.

Manhattan distance

$$d(x_i, x_j) = \sum_{k=1}^p |x_{ik} - x_{jk}|$$



Quantitative variables

Manhattan distance also called **city block distance**.

Mahalanobi distance

$$d(x_i, x_j) = \sqrt{(x_i - x_j)^T \Sigma^{-1} (x_i - x_j)}$$

The distance is based on data itself and the two points are assumed to be of the same distribution with equal dispersion Σ .

Quantitative variables

Tanimoto distance

$$d(x_i, x_j) = \frac{x_i^T x_j}{x_i^T x_i + x_j^T x_j - x_i^T x_j}$$
$$= \text{sum}(x_i \ \& \ x_j) \ / \ \text{sum}(x_i \ | \ x_j)$$

Let the sample x have $x_k = 1$ if it possesses the i^{th} attribute, and $x_k = 0$ otherwise.

The ratio of the number of shared attributes to the number possessed by x_i and x_j .

Often used in information retrieval and biological taxonomy - works well for categorical variables.

Weighted distances

$$d(x_i, x_j) = \sum_{k=1}^p w_k d(x_{ik}, x_{jk}),$$

$$\sum_{k=1}^p w_k = 1$$

Give different weight to the p attributes (variables).

Note that setting $w_k = 1/p$ does *not* necessarily give equal influence to the attributes.

We would have to normalize with the average distance for the k^{th} attribute.

Unsupervised clustering

We have data but no information about class belonging

- ▶ Group data in clusters
- ▶ Observations that are "near" each other should belong to the same class/cluster
- ▶ This can help us unveil an unknown structure in data
- ▶ It is like classification but without an answer, ie unsupervised

K-means clustering

- ▶ Super simple algorithm for clustering

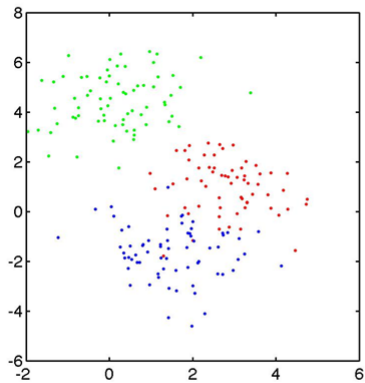
K-means clustering

Decide how many clusters there should be, that is K .

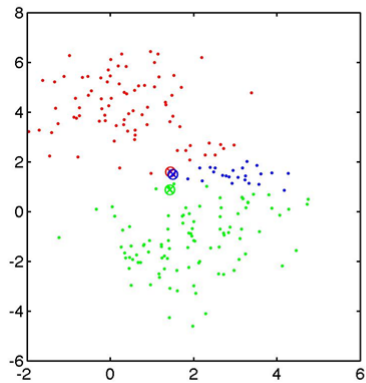
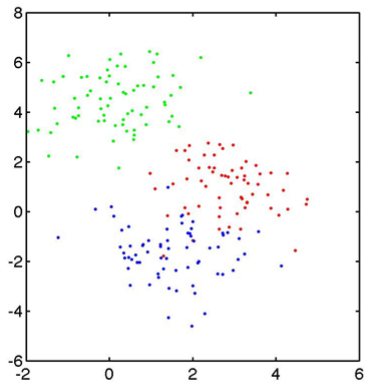
Alternating between two steps (until assignments do not change)

- ▶ Assign each point to the closest cluster center
- ▶ Compute new cluster centers according to the assignments

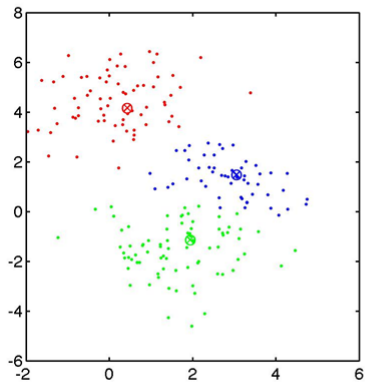
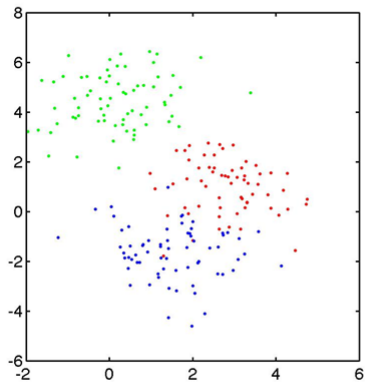
Example



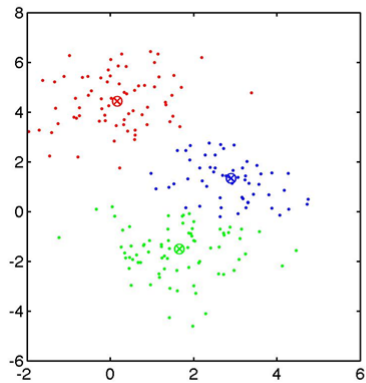
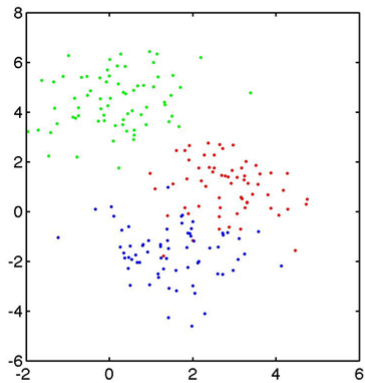
Example



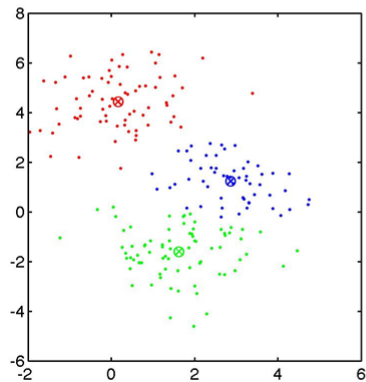
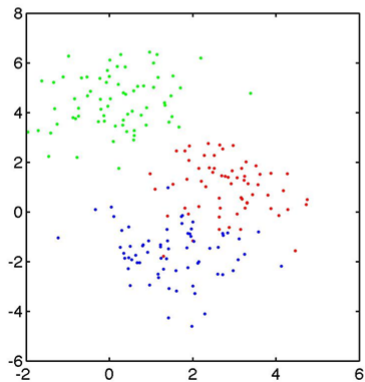
Example



Example



Example



K-medoids

- ▶ K-medoids use one of the observations as cluster center
 - ▶ Computationally much heavier than K-means
- ▶ Makes K-medoids more robust to outliers than K-means
- ▶ Also referred to as PAM, Partitioning Around Medoids

Exercise - K-means clustering

- ▶ Three classes each sampled from a 2D mixture of Gaussian distributions
- ▶ Try different numbers of clusters
- ▶ Kmeans_demo.m



Hierarchical Clustering

- ▶ Generates a tree of observations
- ▶ Each level of the tree reflects one number of clusters.
- ▶ From all data in one cluster down to one observation in each cluster

Hierarchical clustering

- ▶ Do not require input of number of clusters
- ▶ Uses dissimilarity between clusters
- ▶ Two approaches
 - ▶ Bottom-up: Agglomerative (commonly used)
 - ▶ Top-down: Divisive
- ▶ $n - 1$ levels in the hierarchy
- ▶ At each level perform split or merge which gives largest between-group dissimilarity

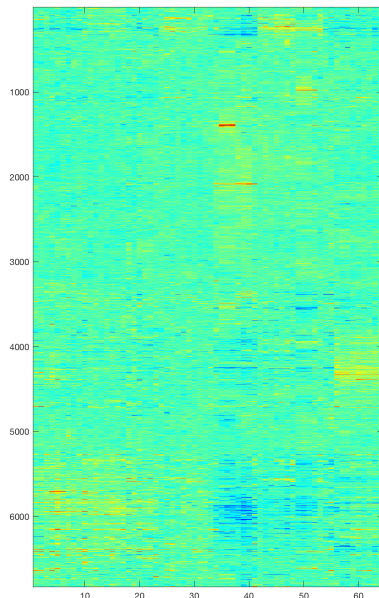
Microarray example

- ▶ The mRNA microarray data from ESL, example 4
- ▶ Samples from cancer tumors
- ▶ 6830 genes
- ▶ 64 samples

Unsupervised clustering:

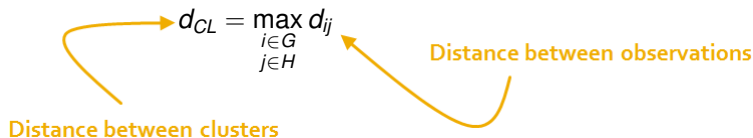
Which samples are most similar to each other, in terms of their expression profiles across genes.

Do similar samples share the same form of cancer?



Complete-linkage hierarchical clustering

Cluster-cluster distance measured as the distance from the furthest pair of points (i, j) from the clusters (G, H) respectively.



The diagram illustrates the concept of complete-linkage clustering. It features the equation $d_{CL} = \max_{\substack{i \in G \\ j \in H}} d_{ij}$ in the center. Two orange arrows originate from this equation. One arrow curves downwards and to the left, pointing towards the text "Distance between clusters". The other arrow curves downwards and to the right, pointing towards the text "Distance between observations".

$$d_{CL} = \max_{\substack{i \in G \\ j \in H}} d_{ij}$$

Distance between clusters

Distance between observations

Also called the furthest-neighbor technique

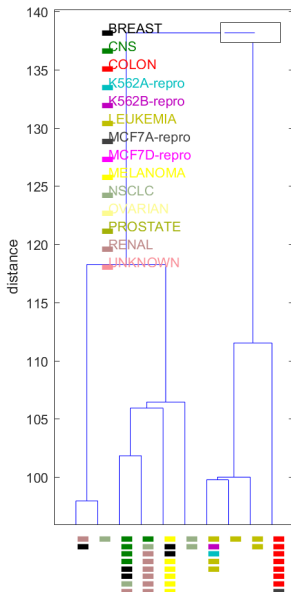
Microarray example - complete-linkage clustering

- ▶ Sample-sample: Euclidean
- ▶ Cluster-cluster: Complete
- ▶ Dendrogram cut at 10 nodes

Tends to give balanced trees independent of data

A **dendrogram** consists of many U-shaped lines connecting objects in a hierarchical tree.

The **height** of each U represents the distance between the two objects being connected.



Single-linkage hierarchical clustering

Cluster-cluster distance measured as the distance of the closest pair of points (i, j) from the clusters (G, H) respectively.

$$d_{SL} = \min_{\substack{i \in G \\ j \in H}} d_{ij}$$

Also called the nearest-neighbor technique

Chaining - a problem

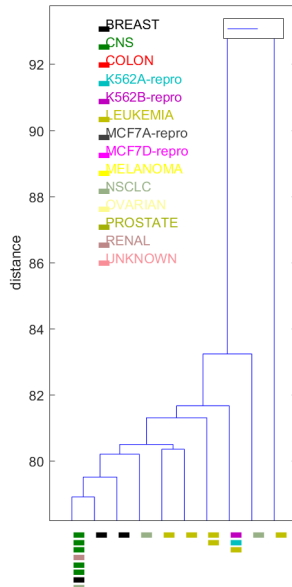
Microarray example - single-linkage clustering

- ▶ Sample-sample: Euclidean
- ▶ Cluster-cluster: Single

Tends to give unequal cluster sizes

In Matlab:

```
% Sample-sample distance matrix  
Y = pdist(X,'euclidean');  
  
% Create hierarchical cluster  
Z = linkage(Y,'single');  
  
% Dendrogram plot  
dendrogram(Z)
```



Ward-linkage hierarchical clustering

Cluster-cluster distance measured as the increment in within-cluster sum of squares

$$d_{Ward} = \sqrt{n_G n_H \frac{\|\bar{x}_G - \bar{x}_H\|_2^2}{(n_G + n_H)}}$$

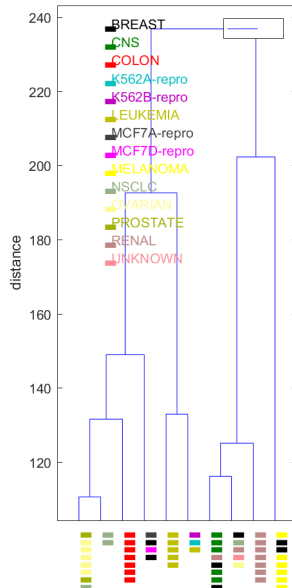
Scaled version of the centroid-linkage distance. Ward's distance measure how much the sum of squares will increase when we merge the two clusters,

$$d_{Ward} = \sum_{i \in G \cup H} \|x_i - \bar{x}_{G \cup H}\|_2^2 - \sum_{i \in G} \|x_i - \bar{x}_G\|_2^2 - \sum_{i \in H} \|x_i - \bar{x}_H\|_2^2$$

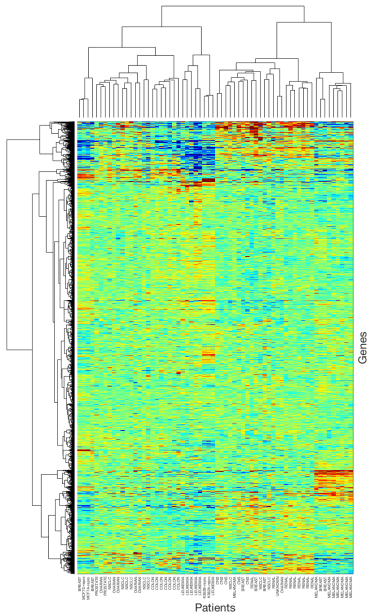
Microarray example - Ward-linkage clustering

- ▶ Sample-sample: Euclidean
- ▶ Cluster-cluster: Ward

Tends to give a good compromise between balanced/unbalanced clusters



Two way clustering



- ▶ Cluster both genes and patients
- ▶ Presented as a heatmap

In Matlab:
`clustergram(X)`

A warning...

Hierarchical clustering will **always** generate a dendrogram

- ▶ Even when data is completely random
- ▶ Be careful with the interpretation
- ▶ It is the application that tells if the structure is relevant

Exercise - Hierarchical clustering

- ▶ Zip-data - example 3 in ESL
 - ▶ 400 samples of handwritten digits 0-9
 - ▶ 16×16 grayscale images = 256 features
-
- ▶ Try different dissimilarity measures
 - ▶ Where would you cut the dendrogram (ie how many clusters should we choose)?
 - ▶ HierarchicalEx.m



The problem of validation

- ▶ How to select the number of clusters?

Selecting the number of clusters

- ▶ Gap-statistics
- ▶ Goodness-of-fit measures
 - ▶ Chi-squared statistics
 - ▶ Kolmogorov-Smirnov statistics
 - ▶ AIC and BIC
- ▶ Biological or physical interpretation



Why not cross validation?

Within cluster dissimilarity

Distance between all points in one cluster (Euclidean distance)

$$\begin{aligned} D_\ell &= \sum_{\substack{C_k(i)=\ell \\ C_k(j)=\ell}} ||x_i - x_j||^2 \\ &= \dots \\ &= 2N_\ell \sum_{C_k(i)=\ell} ||x_i - \bar{x}_\ell||^2 \end{aligned}$$

Within cluster dissimilarity

$$W_k = \sum_{\ell} \frac{1}{2N_\ell} D_\ell$$

($C_k(i)$ = cluster for obs. i when we have k clusters in the model.)

Gap-statistic

Compares the log criterion value with K clusters to the expected log criterion value for **uniformly distributed** data (20 simulations)

$$G(K) = \log(U_k) - \log(W_k)$$

- ▶ U_k within cluster dissimilarity, simulated data - mean over 20 samples
- ▶ W_k within cluster dissimilarity, actual data

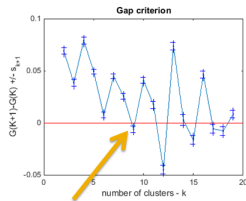
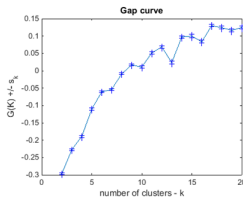
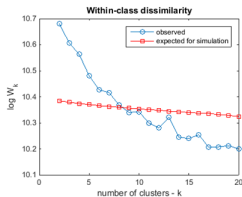
Choose

$$K^* = \arg \min_k \{K | G(K) \geq G(K+1) - s'_{K+1}\}$$

where

$$s'_{K+1} = \text{std}(\log(U_K))\sqrt{1 + 1/20}$$

Example



Optimal $K^* = 9$

Based on simulation - might differ from one simulation to another

Gap-statistic

- ▶ Can be used both for K-means, K-medoids and hierarchical clustering
- ▶ Works with different measures of the within-cluster dissimilarity

Exercise - Gap-statistics for K-means clustering

- ▶ K-means clustering on the zip data
- ▶ Including calculation of within-class dissimilarities and gap statistic
- ▶ KmeansEx.m
- ▶ Try first `kmeans` and then `kmedoids`



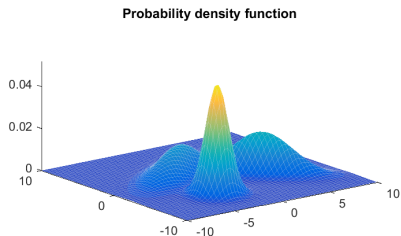
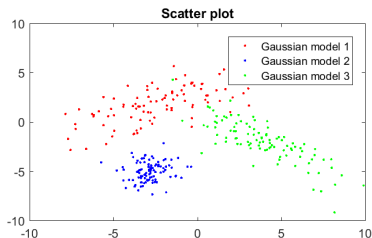
Gaussian Mixture Modeling and Expectation Maximization

Gaussian Mixture Modeling:

- ▶ Data belongs to one of several Gaussian distributions
- ▶ An un-observed (latent) random variable selects which distribution the observation comes from.
- ▶ This gives a complicated likelihood function
- ▶ Easily solved using the EM algorithm



Gaussian models



Observations from three
Gaussian distributions

$$\begin{cases} X_i \in N(\mu_1, \Sigma_1) & \text{if } Z_i = 1 \\ X_i \in N(\mu_2, \Sigma_2) & \text{if } Z_i = 2 \\ X_i \in N(\mu_3, \Sigma_3) & \text{if } Z_i = 3 \end{cases}$$

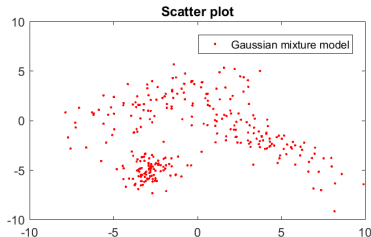
Known variables, Z_i , indicate
cluster/distribution

$$\begin{cases} Z_i = 1 & i = 1, \dots, 100 \\ Z_i = 2 & i = 101, \dots, 200 \\ Z_i = 3 & i = 201, \dots, 300 \end{cases}$$

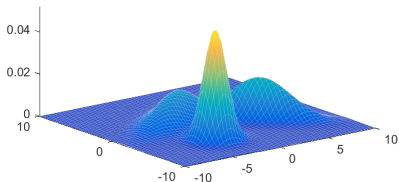


What are the unknown model
parameters?

Gaussian mixture models



Probability density function



Observations from three Gaussian distributions

$$\begin{cases} X_i \in N(\mu_1, \Sigma_1) & \text{if } Z_i = 1 \\ X_i \in N(\mu_2, \Sigma_2) & \text{if } Z_i = 2 \\ X_i \in N(\mu_3, \Sigma_3) & \text{if } Z_i = 3 \end{cases}$$

Un-observed variables, Z_i , indicate cluster/distribution

$$\begin{cases} P(Z_i = 1) = \tau_1 \\ P(Z_i = 2) = \tau_2 \\ P(Z_i = 3) = \tau_3 \end{cases}$$

The probabilities sum to one,

$$\tau_1 + \tau_2 + \tau_3 = 1$$



What are the unknown model parameters?

Likelihood for the Gaussian mixture example

Parameters in the Gaussian mixture model are,

$$\theta = (\tau_1, \tau_2, \tau_3, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3)$$

and

$$\mathbf{Z} = (Z_1, \dots, Z_n) \quad (n \text{ might be very large})$$

The likelihood is

$$l(\theta; \mathbf{x}, \mathbf{Z}) = \prod_{i=1}^n \sum_{j=1}^3 \mathbb{I}_{\{Z_i=j\}} \tau_j f(x_i; \mu_j, \Sigma_j)$$

with ML estimate

$$\theta_{ML} = \arg \max_{\theta, \mathbf{Z}} \log l(\theta; \mathbf{x}, \mathbf{Z})$$

Finding a solution (θ, \mathbf{Z}) is much simplified by the **EM-algorithm**.

The EM-algorithm

The algorithm is a two-step iteration.

Expectation step: Define the expectation value,

$$Q(\theta|\theta^k) = E_{Z|x, \theta^k} L(\theta; x, Z)$$

Maximization step: Find parameter estimate,

$$\theta^{k+1} = \arg \max_{\theta} Q(\theta|\theta^k)$$

First step defines the expectation value of the log likelihood given observed data, x , and current value of the parameter estimate, θ^k .

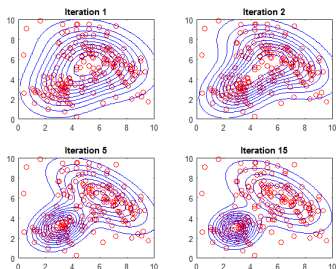
Second step chooses optimal θ given the expectation value whereupon the procedure is repeated until convergence.

- Problems with local minimas.

The Gaussian mixture EM-algorithm

1. Initialize means μ , covariances, Σ and mixing coefficients τ .
2. **Expectation step**
Calculate conditional probabilities T_{ij} for Z_i belonging to cluster j using Bayes formula.
3. **Maximization step**
Calculate weighted (using T_{ij}) mean and covariance estimates μ Σ . Calculate mixing coefficients τ based on mean of weights.
4. Iterate until convergence

Gaussian mixture in Matlab



Matlab simulation of a Gaussian Mixture model

$$P(Z_i = 1) = 0.3$$

$$P(Z_i = 2) = 0.7$$

$$X_i \in N\left(\begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\right) \quad \text{if } Z_i = 1$$

$$X_i \in N\left(\begin{bmatrix} 6 \\ 6 \end{bmatrix}, \begin{bmatrix} 4 & -2 \\ -2 & 4 \end{bmatrix}\right) \quad \text{if } Z_i = 2$$

and $i = 1 \dots 200$

Some tricks

- ▶ High dimension (large p) and few data (small n)
 - ▶ Postulate same covariance structure for all clusters,
 $\Sigma = \Sigma_1 = \dots = \Sigma_K$
 - ▶ Postulate diagonal covariance structure, $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_K)$
 - ▶ Regularize, $\Sigma = \Sigma + \lambda I$
 - ▶ Use first principal components instead of all dimensions in data
- ▶ Selecting number of clusters K
 - ▶ Clustering based on Gaussian model
 - ▶ Use information criteria (AIC or BIC) to select number of clusters
- ▶ The structure in data is described by μ , Σ and τ

Exercise - Gaussian mixture

We have data with four different measures from flowers of three different species (Fisheriris.mat). There are 50 observations of each species. See if you can identify three clusters in data using gaussian mixture modelling. (Two of the species are very similar)



Iris setosa



Iris versicolor



Iris virginica

- ▶ Plot data using `plotmatrix`
- ▶ Loop over different numbers of clusters
- ▶ Plot BIC/AIC for different model orders
- ▶ Notice the different extra parameters to the `gmdistribution.fit` function - they might be necessary

