第6-3章: Clustering Analysis

1. Hierarchical clustering
2. Model-based clustering

References:

- M. Eisen et al.: Cluster analysis and display of genome-wide expression patterns. Proc.Natl.Acad.Sci.USA 95, 14863-8, 1998
- Wei Pan, Jizhen Lin and Chap T Le. Model-based cluster analysis of microarray gene-expression data. Genome Biology 3(2): research0009.1–0009.8, 2002.
- G.J. McLachlan, R.W. Bean, and D. Peel, A Mixture Model-Based Approach to the Clustering of Microarray Expression Data. Bioinformatics 18, 413-422, 2002.

- Eisen MB, Spellman PT, Brown PO, Botstein D. 1998. Cluster analysis and display of genome-wide expression patterns. *PNAS* 95: 14863-14868.

- Google scholar citation: 13061 (04/25/2013), 13066(04/27/2013)

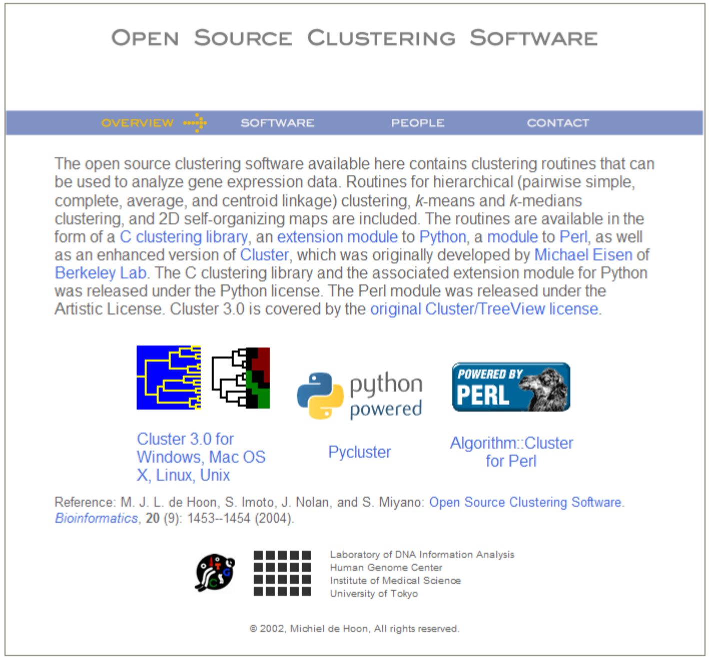
Cluster Analysis and Visualization Software

- Cluster 3.0

http://bonsai.hgc.jp/~mdehoon/software/cluster/software.htm

- TreeView

http://www.eisenlab.org/eisen/?page_id=42



Open Source Clustering Software

The open source clustering software available here contains clustering routines that can be used to analyze gene expression data. Routines for hierarchical (pairwise simple, complete, average, and centroid linkage) clustering, *k*-means and *k*-medians clustering, and 2D self-organizing maps are included. The routines are available in the form of a C clustering library, an extension module to Python, a module to Perl, as well as an enhanced version of Cluster, which was originally developed by Michael Eisen of Berkeley Lab. The C clustering library and the associated extension module for Python was released under the Python license. The Perl module was released under the Artistic License. Cluster 3.0 is covered by the original Cluster/TreeView license.

Cluster 3.0 for Windows, Mac OS X, Linux, Unix

Pycluster

Algorithm::Cluster for Perl

Reference: M. J. L. de Hoon, S. Imoto, J. Nolan, and S. Miyano: Open Source Clustering Software. *Bioinformatics*, **20** (9): 1453--1454 (2004).

Laboratory of DNA Information Analysis
Human Genome Center
Institute of Medical Science
University of Tokyo

Maple Tree

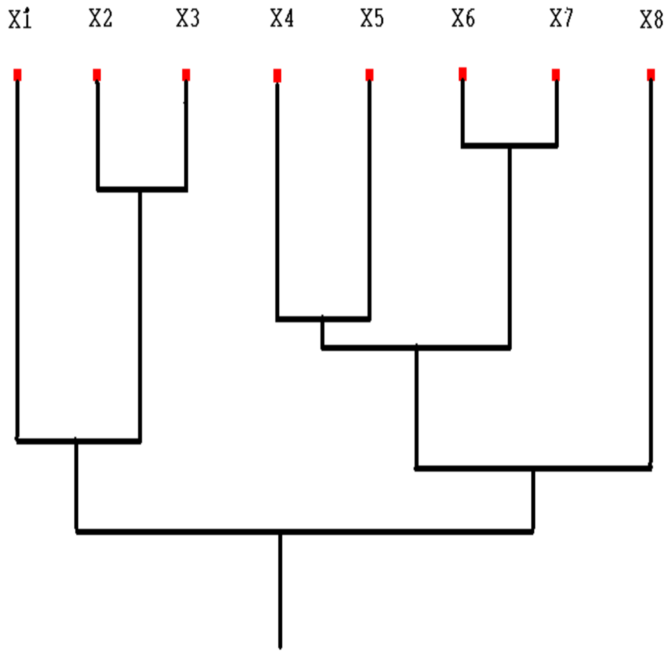
Maple Tree is an open source, cross-platform, visualization tool to graphically browse results of clustering and other analyses from Michael Eisen' Cluster and Aerie. Maple Tree may also be used to visualize results from Michiel Jan Laurens de Hoon and Sunyong Kim's version of Cluster.

Maple Tree is intended to be an alternative to Michael Eisen's TreeView, and is being developed in conjunction with his lab at the Lawrence Berke National Laboratory. As new analyses become available as part of Aerie, uniquely tailored visualizations will be added to Maple Tree.

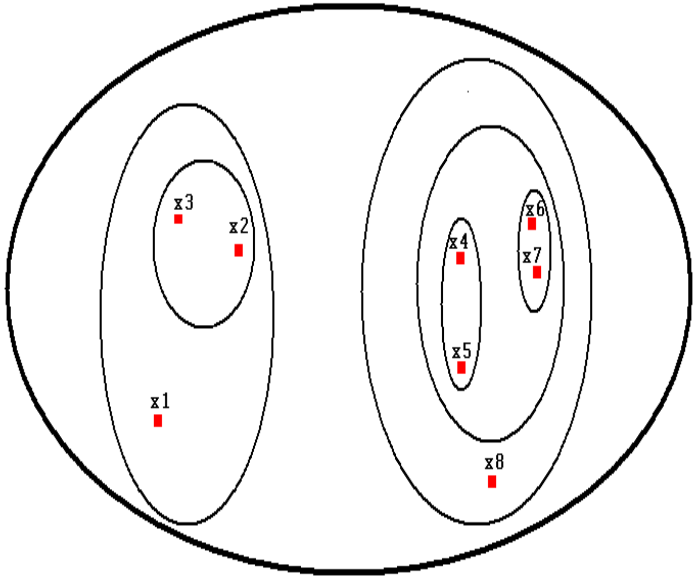
Visit our SourceForge site to download releases, file bug reports, and subscribe to one of our mailing lists.

http://mapletree.sourceforge.net/

Hierarchical Clustering



Dendrogram



Venn Diagram of Clustered Data

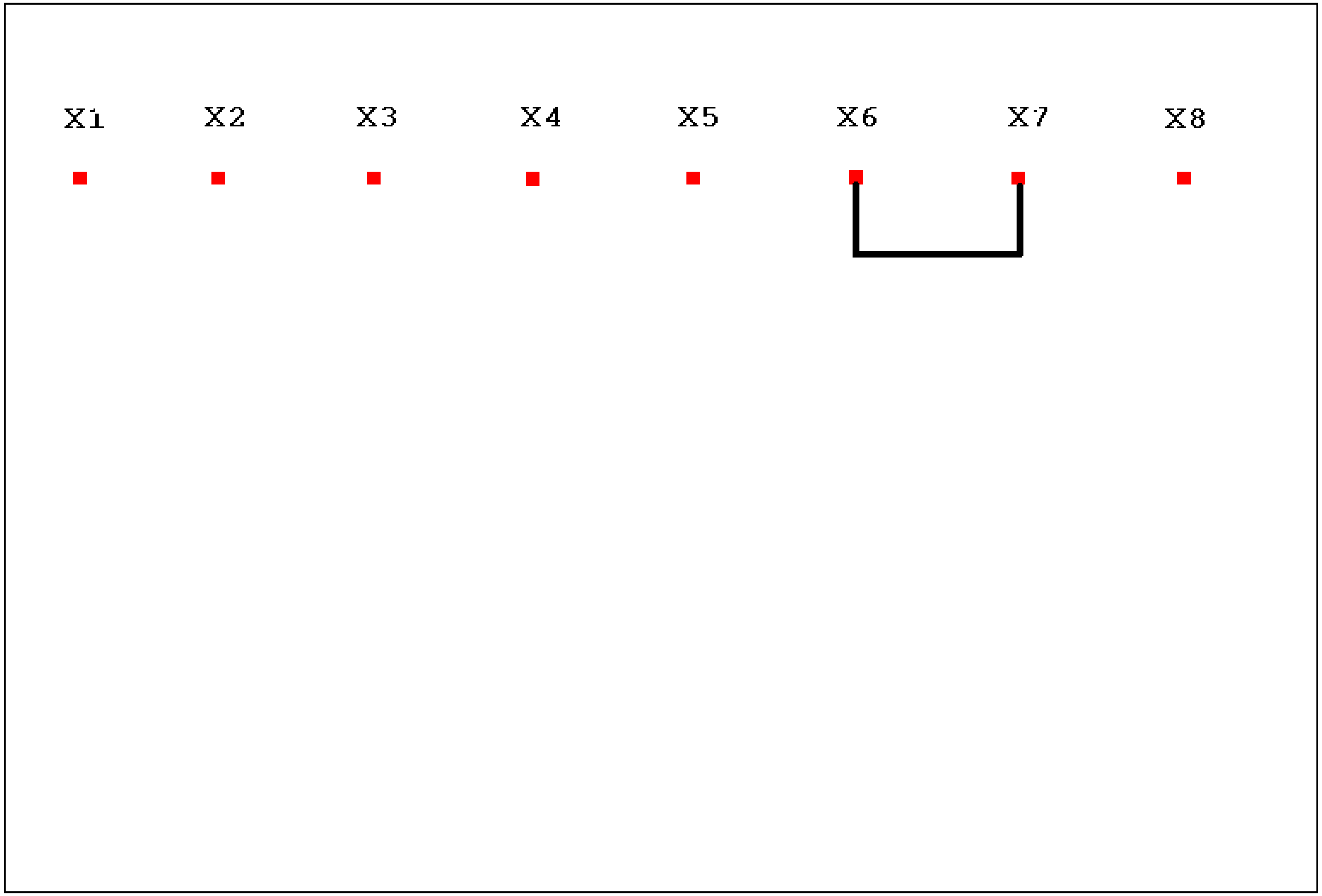
From http://www.stat.unc.edu/postscript/papers/marron/Stat321FDA/RimaIzempresentation.ppt

Nearest Neighbor Algorithm

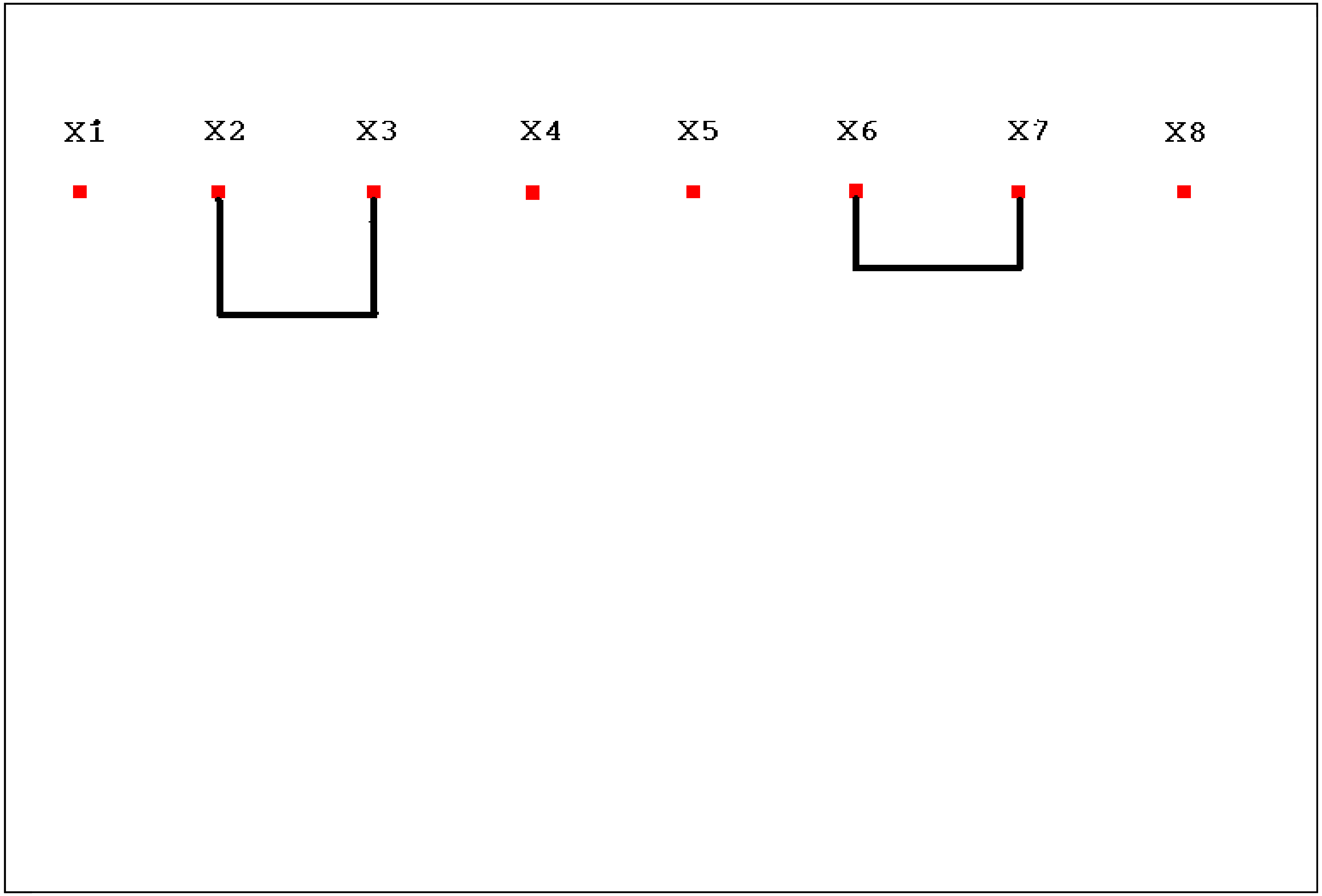
- Nearest Neighbor Algorithm is an agglomerative approach (bottom-up).

- Starts with *n* nodes (*n* is the size of our sample), merges the 2 most similar nodes at each step, and stops when the desired number of clusters is reached.

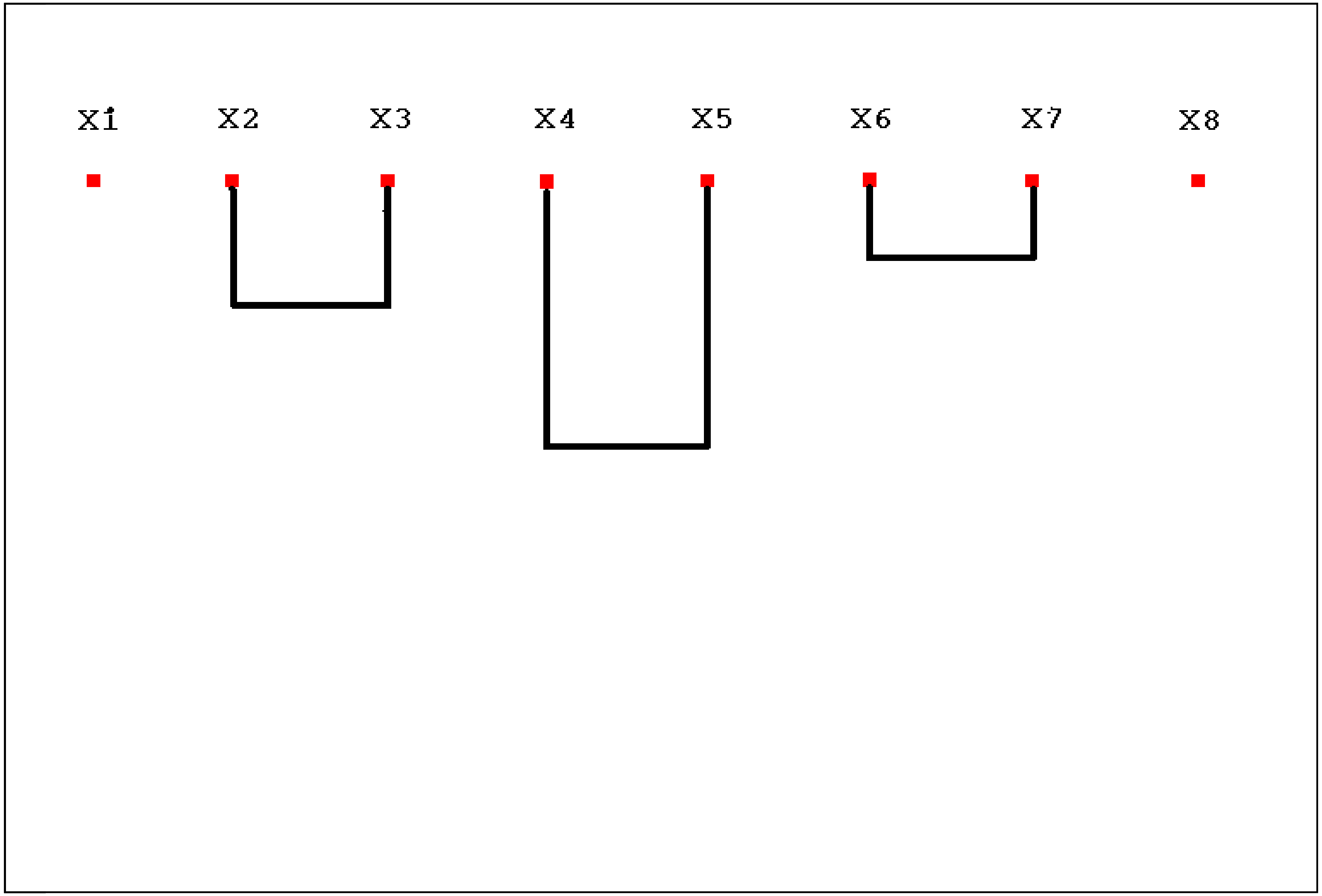
From http://www.stat.unc.edu/postscript/papers/marron/Stat321FDA/RimaIzempresentation.ppt

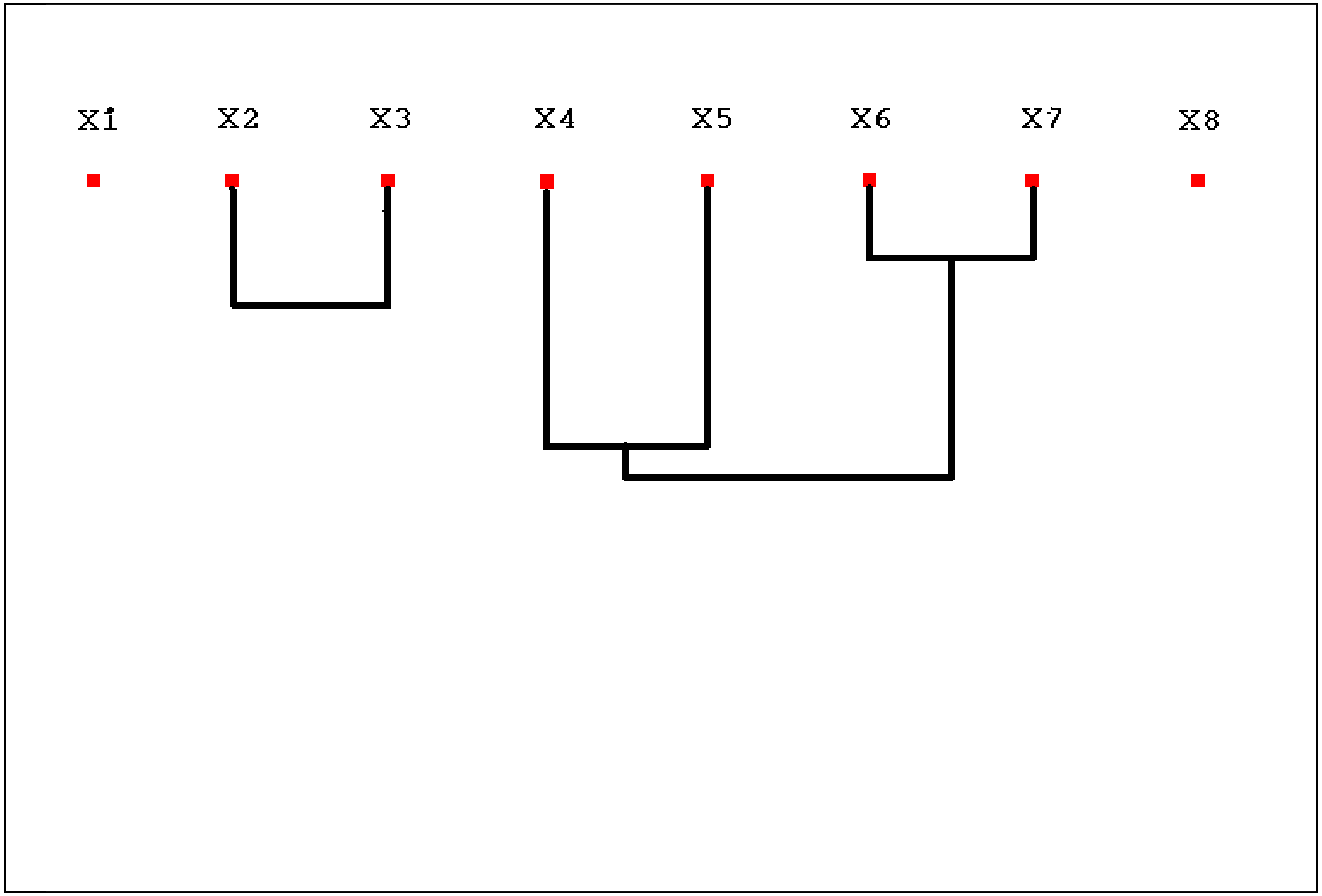


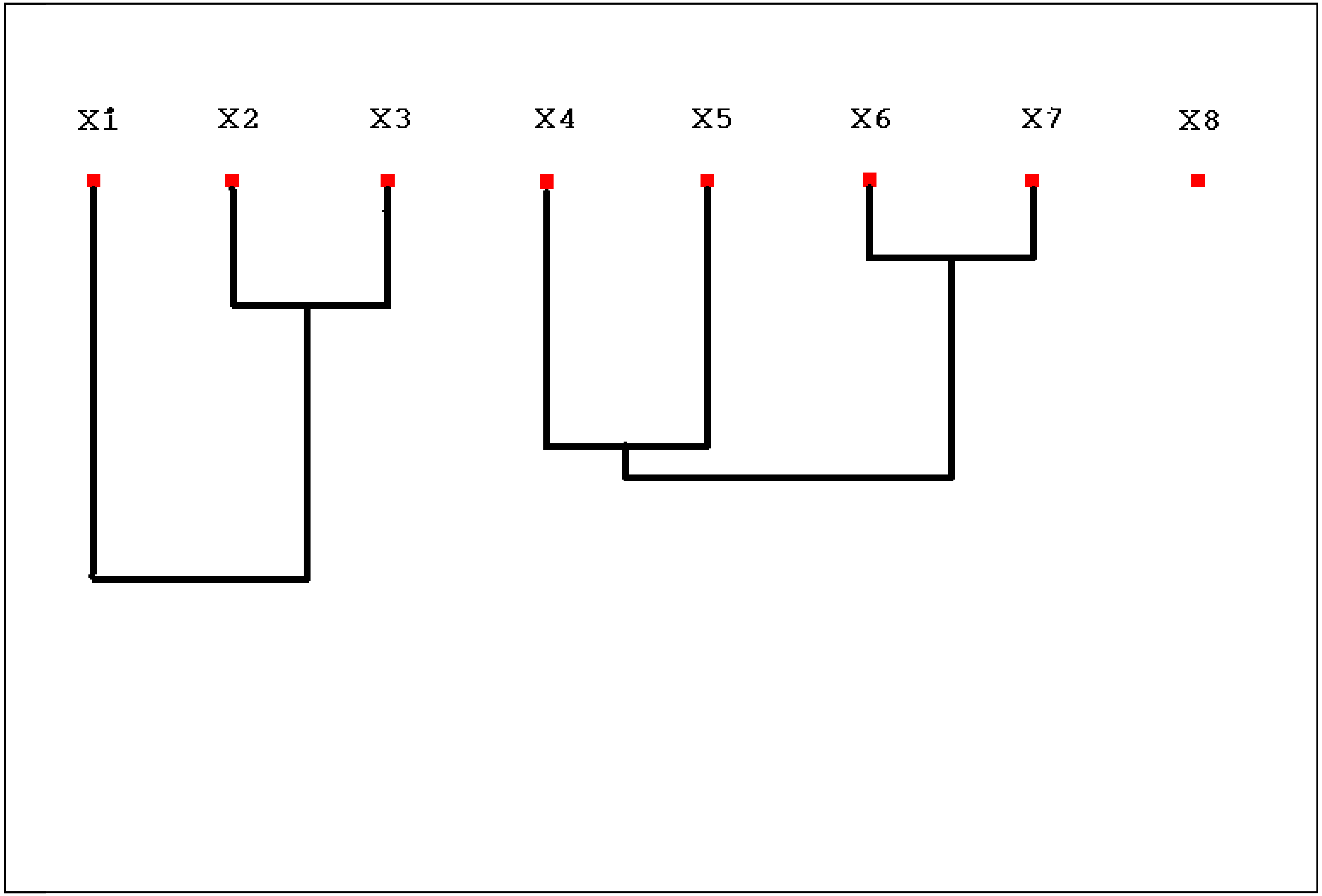
From http://www.stat.unc.edu/postscript/papers/marron/Stat321FDA/RimaIzempresentation.ppt

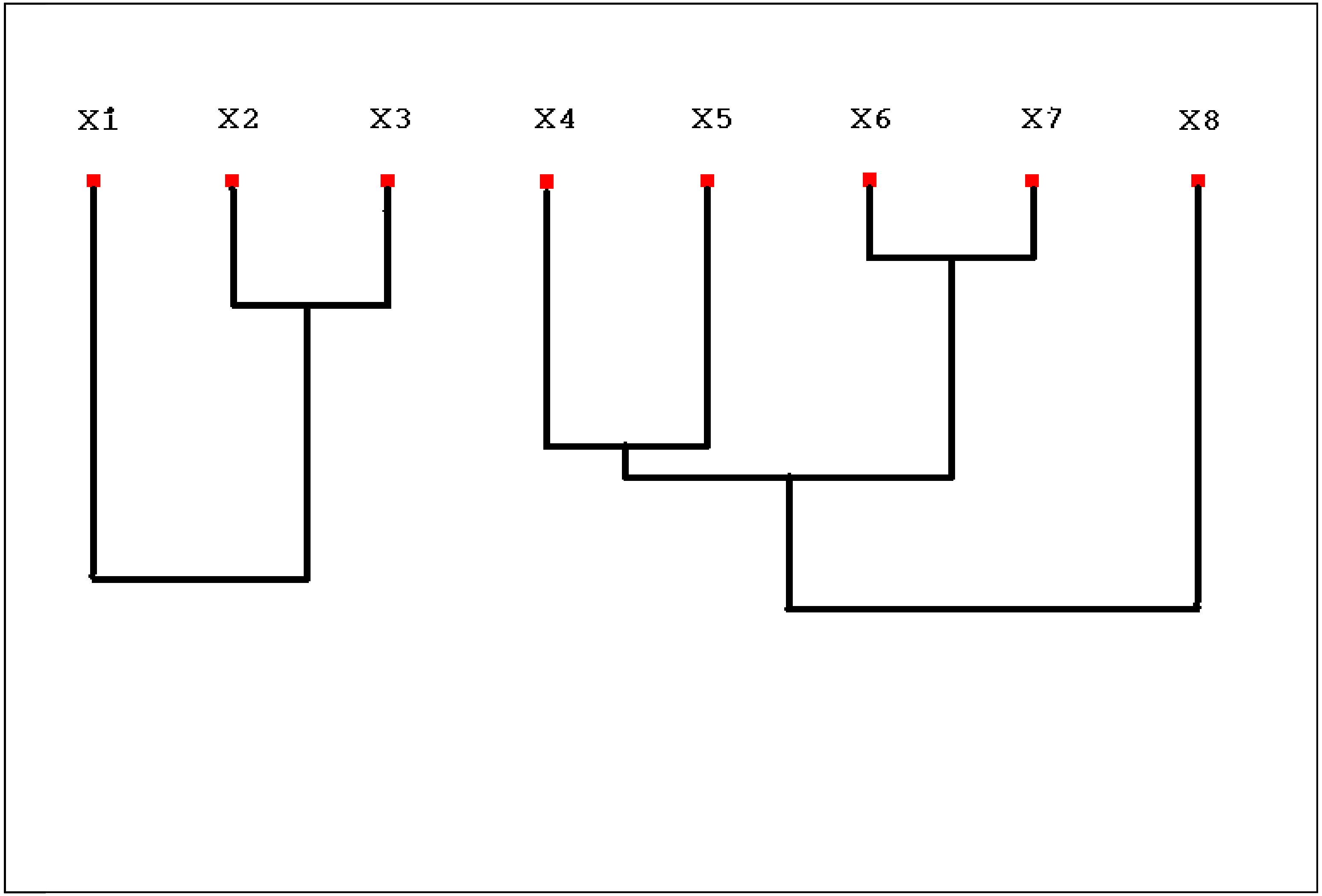


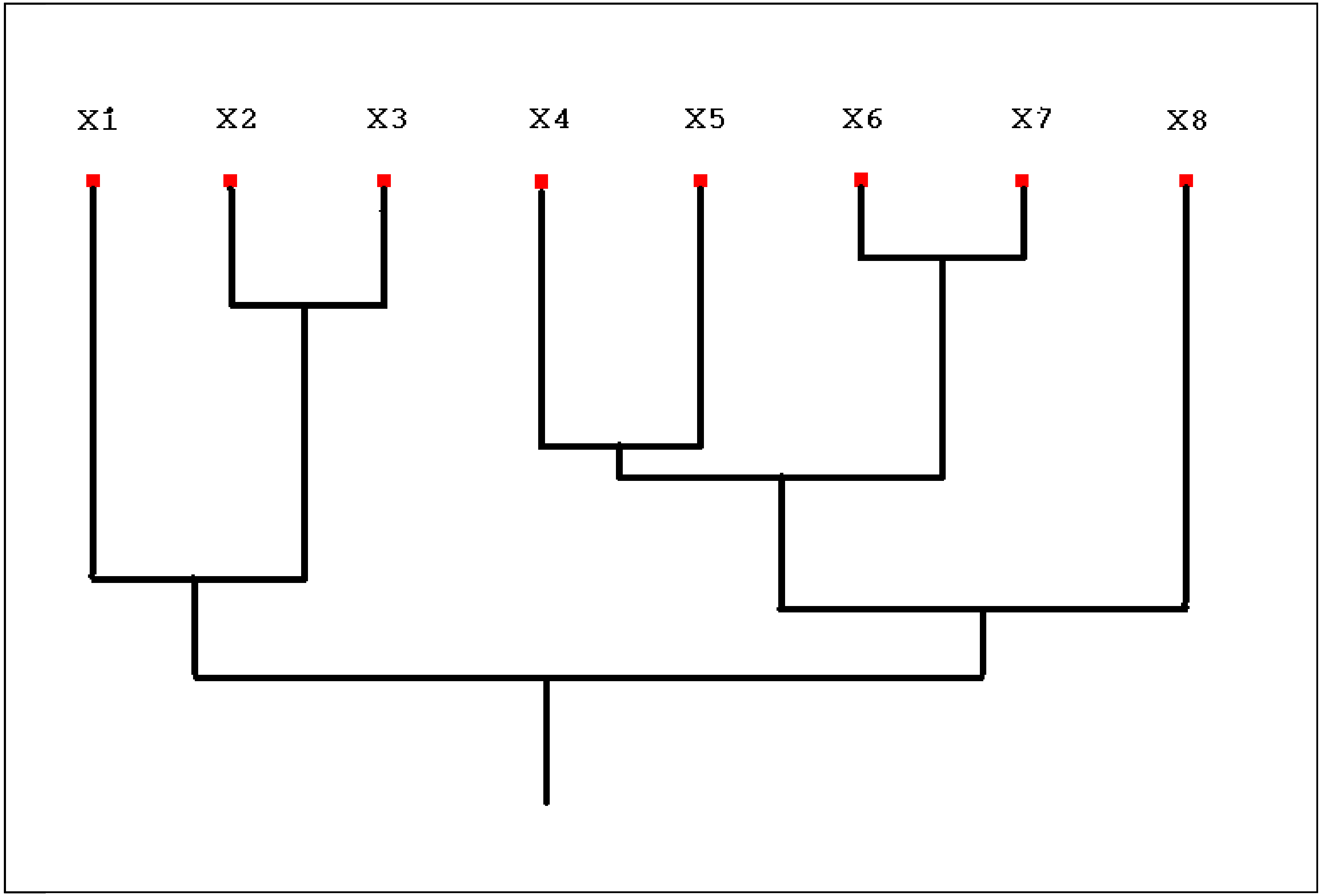
X1
X2
X3
X4
X5
X6
X7
X8


X1
X2
X3
X4
X5
X6
X7
X8


X1
X2
X3
X4
X5
X6
X7
X8


X1
X2
X3
X4
X5
X6
X7
X8


X1
X2
X3
X4
X5
X6
X7
X8


X1
X2
X3
X4
X5
X6
X7
X8

Similarity Measurements

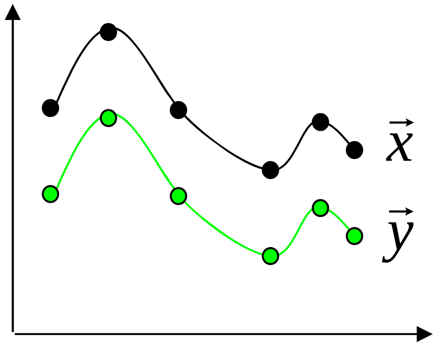
- Pearson Correlation

Two profiles (vectors) $\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$ and $\vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$

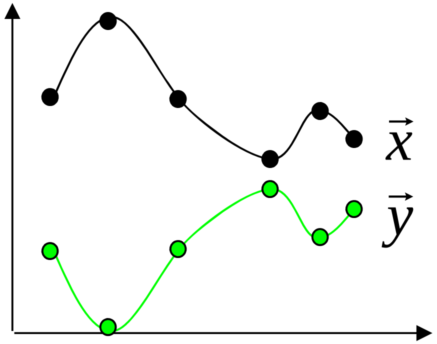
$$C_{pearson}(\vec{x},\vec{y}) = \frac{\sum_{i=1}^{N}(x_i - m_x)(y_i - m_y)}{\sqrt{[\sum_{i=1}^{N}(x_i - m_x)^2][\sum_{i=1}^{N}(y_i - m_y)^2]}}$$

$$m_x = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$m_y = \frac{1}{N}\sum_{n=1}^{N} y_n$$



$+1 \geq$ Pearson Correlation ≥ -1



Similarity Measurements

- Euclidean Distance

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \quad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{n=1}^{N} (x_n - y_n)^2}$$

Similarity Measurements

- Cosine Correlation

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$$C_{\text{cosine}}(\vec{x}, \vec{y}) = \frac{\frac{1}{N}\sum_{i=1}^{N} x_i \times y_i}{\|\vec{x}\| \times \|\vec{y}\|}$$

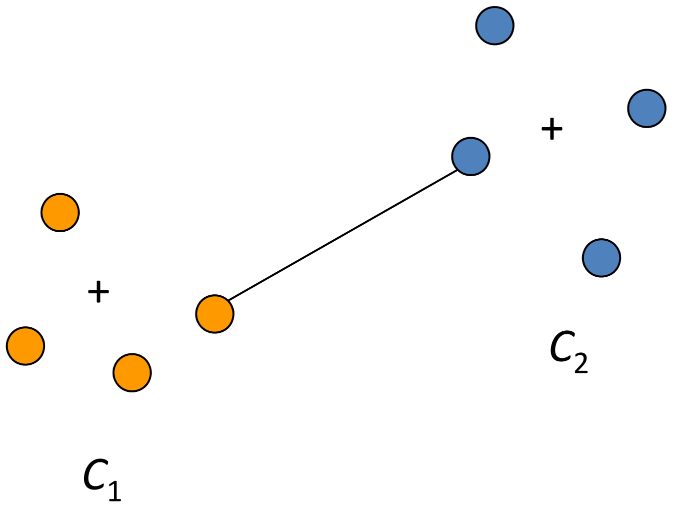
$\vec{x} = \vec{y}$ $\qquad$ $+1 \geq$ Cosine Correlation ≥ -1 $\qquad$ $\vec{x} = -\vec{y}$

Group Similarity

- Single linkage
- Complete linkage
- Average linkage
- Average group linkage

Clustering

Single Linkage

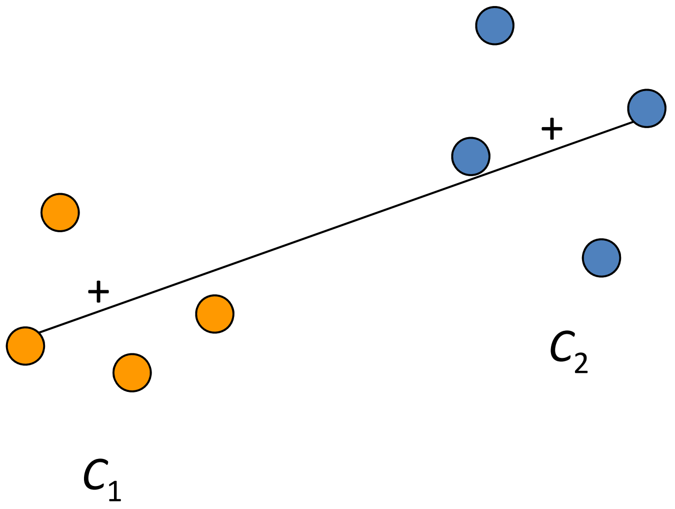


Dissimilarity between two clusters = Minimum dissimilarity between the members of two clusters

Tend to generate "long chains"

Clustering

Complete Linkage

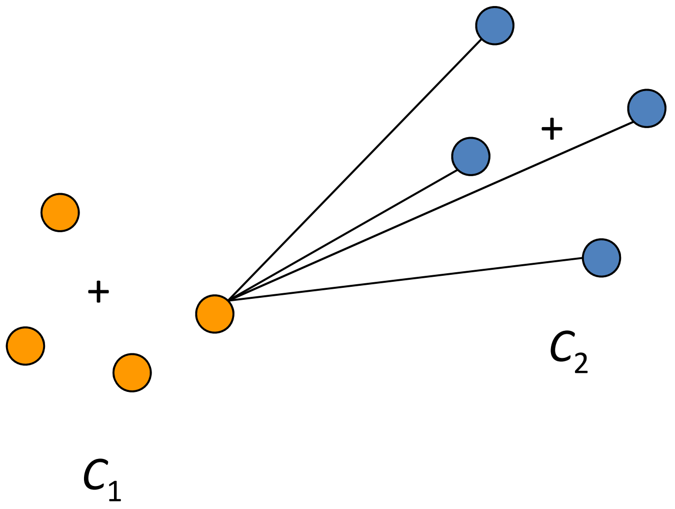


Dissimilarity between two clusters = Maximum dissimilarity between the members of two clusters

Tend to generate “clumps”

Clustering

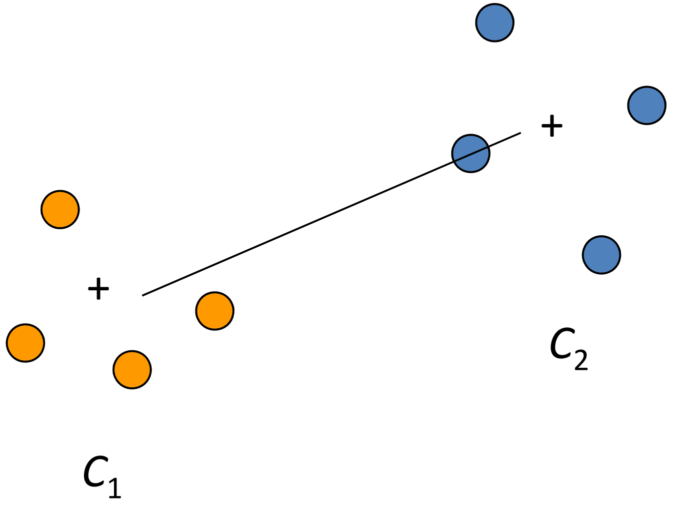
Average Linkage



Dissimilarity between two clusters = Averaged distances of all pairs of objects (one from each cluster).

Clustering

Average Group Linkage



Dissimilarity between two clusters =
Distance between two cluster means.

Other Clustering Methods

- K-means, fuzzy k-means
- Self-organization mapping (SOM)
- Gaussian mixture model, Bayesian clustering algorithms
- Nonnegative Matrix factorization
- Iterative signature algorithm (ISA), progressive iterative signature algorithm (PISA)…
- Biclustering

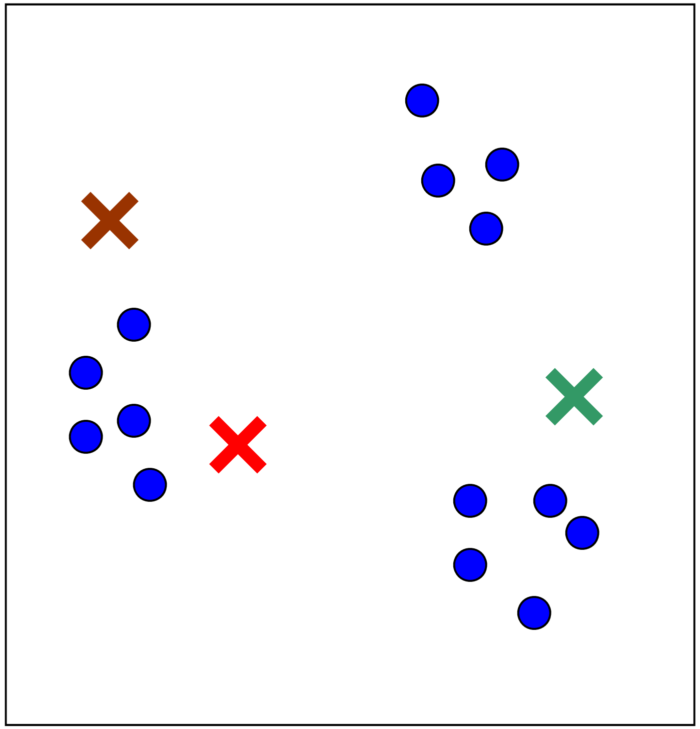
K-means Algorithm

1. Choose K centroids at random
2. Make initial partition of objects into k clusters by assigning objects to closest centroid
3. Calculate the centroid (mean) of each of the k clusters.
4. a. For object i, calculate its distance to each of the centroids.

   b. Allocate object i to cluster with closest centroid.

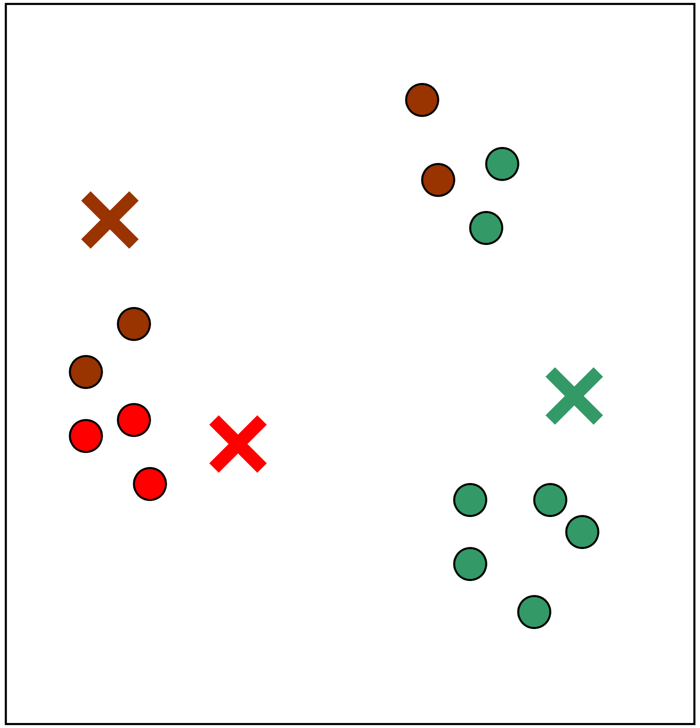
   c. If object was reallocated, recalculate centroids based on new clusters.
4. Repeat 3 for object i = 1,….N.
5. Repeat 3 and 4 until no reallocations occur.
6. Assess cluster structure for fit and stability

K-means Algorithm



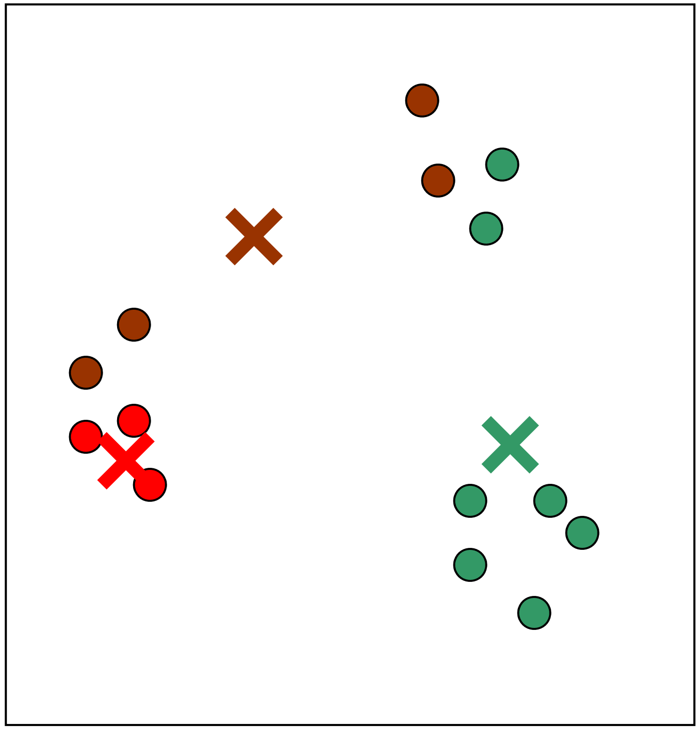
Iteration = 0

K-means Algorithm



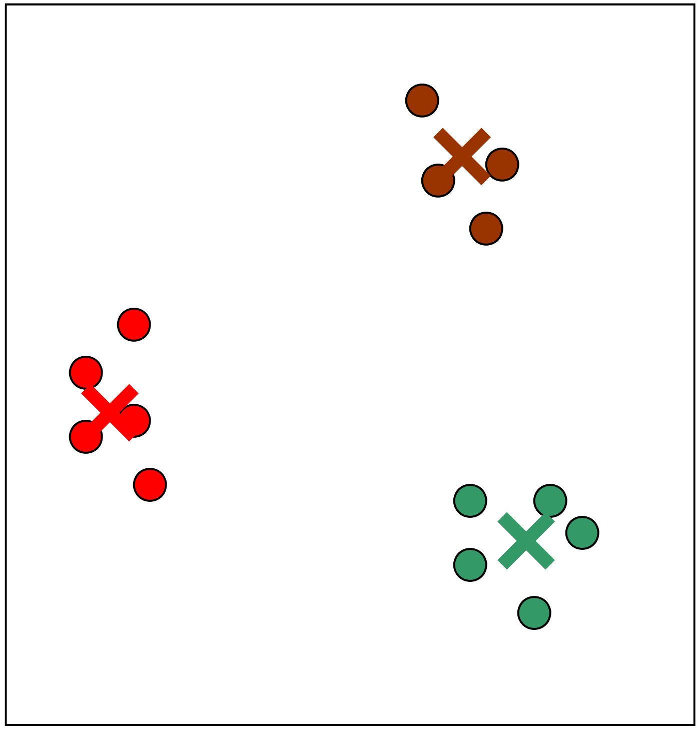
Iteration = 1

K-means Algorithm



Iteration = 2

K-means Algorithm



Iteration = 3

Gaussian Mixture Model

- Each class corresponding to a normal distribution
- The data point y is take to be a realization from a Gaussian mixture model

$$f(y;\Theta) = \sum_{i=1}^{g} \pi_i \phi(y; \mu_i, V_i)$$

Learning the Parameters

- Maximum likelihood estimation. Given data points $y_1, \cdots, y_n$,

$$l(\Theta) = \sum_{i=1}^{n} \log f(y_i; \Theta)$$

- Missing data problem, the class label of each data point.

EM Algorithm

- Iteratively update

$$\begin{cases} \tau_{ij}^{(k)} = \dfrac{\pi_i^{(k)} \phi(y_j; \mu_i^{(k)}, V_i^{(k)})}{f(y_j; \Theta_k)} \\ \pi_i^{(k+1)} = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \tau_{ij}^{(k)} \\ \mu_i^{(k+1)} = \dfrac{\sum_{i=1}^{n} \tau_{ij}^{(k)} y_j}{\sum_{i=1}^{n} \tau_{ij}^{(k)}} \\ V_i^{(k+1)} = \dfrac{\sum_{i=1}^{n} \tau_{ij}^{(k)} (y_j - \mu_i^{(k+1)})(y_j - \mu_i^{(k+1)})^T}{\sum_{i=1}^{n} \tau_{ij}^{(k)}} \end{cases}$$

Choose the Number of Clusters

- Akaike Information Criterion (AIC)

$$AIC = -2l(\hat{\Theta}_g) + 2v_g$$

- Bayesian Information Criterion (BIC)

$$BIC = -2l(\hat{\Theta}_g) + v_g \log(n)$$

where v_g is the number of independent parameters

1. Akaike H: Information theory and an extension of the maximum likelihood principle. In 2nd Int Symp Information Theory. Edited by Petrov BN, Csaki F. Budapest: Akademiai Kiado,1973, 267-281.
2. Schwartz G: Estimating the dimensions of a model. Annls Statistics 1978, 6:461-464.