Cluster Analysis and Visualization

統計與機器學習研討會
Workshop on Statistics and Machine Learning

2004/2/6

吳漢銘 & 陳君厚

Dimension Free Data Visualization

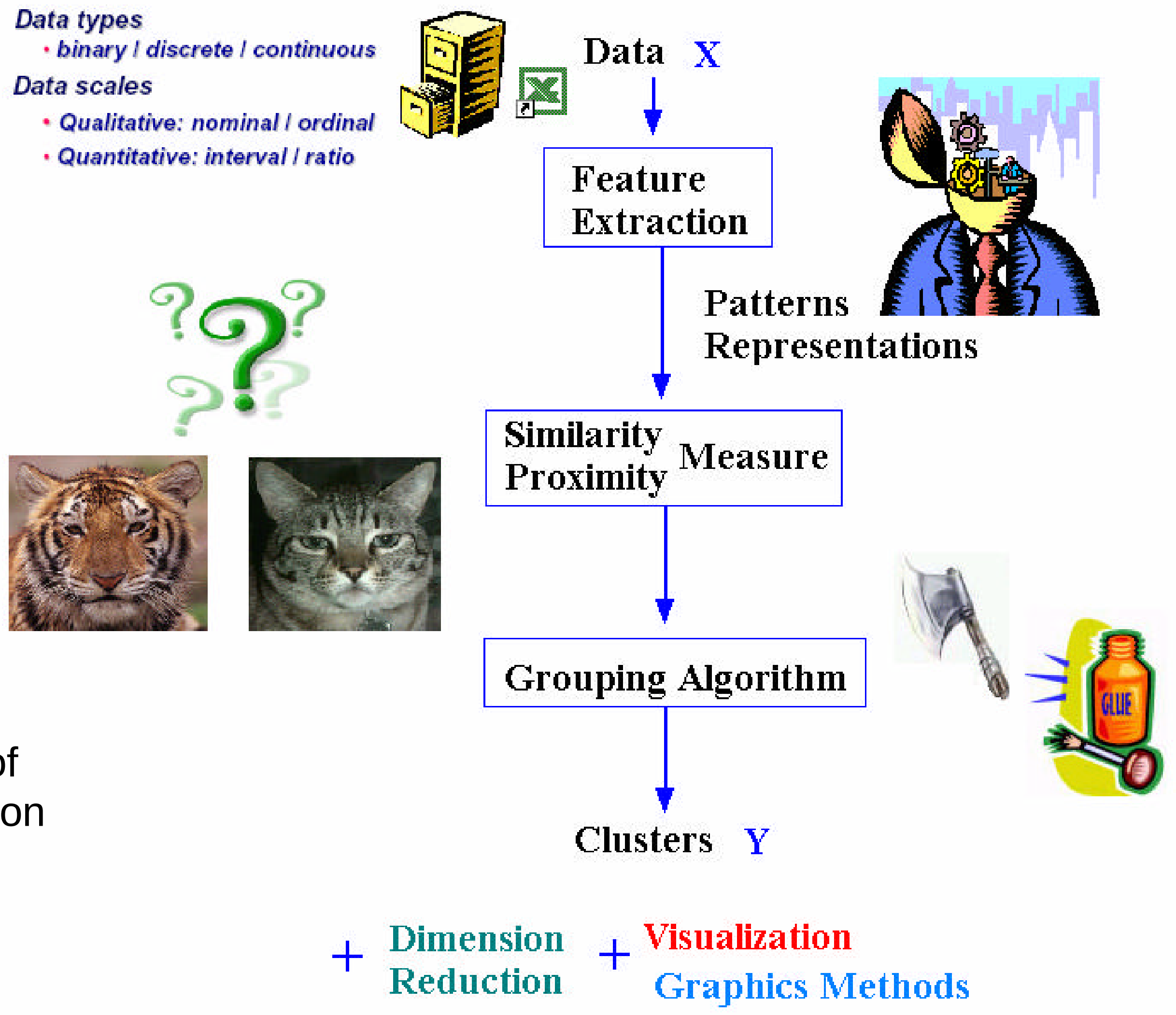
GAP

Institute of Statistical Science, Academia Sinica
中央研究院 統計科學研究所

Outlines

- **Introduction**
  - Stages in Clustering
  - Clustering Analysis and Visualization
- **One/two-dimensional Data**
  - Histogram, Scatterplot, Dendrogram
- **High-dimensional Data : Dimension Reduction Techniques**
  - Principal Component Analysis (PCA)
  - Multidimensional Scaling (MDS)
  - Self-Organizing Maps (SOM)
- **High-dimensional Data: Dimension-free Visualization**
  - Block Clustering
  - Data Image
  - Generalized Association Plots (GAP)

Stages in Clustering



Data types

- *binary / discrete / continuous*

Data scales

- *Qualitative: nominal / ordinal*
- *Quantitative: interval / ratio*

Data X

Feature Extraction

Patterns Representations

Similarity Proximity Measure

Grouping Algorithm

Clusters Y

+ Dimension Reduction + Visualization Graphics Methods

What is clustering?

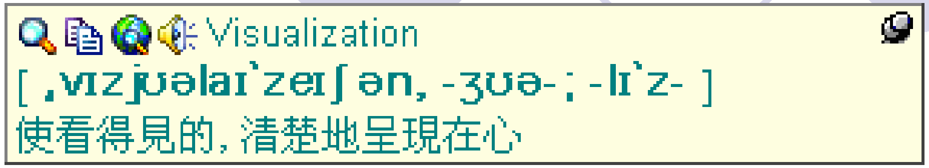
Cluster analysis is the organization of a collection of patterns into clusters based on similarity. The problem is to group a given collection of unlabeled patterns into meaningful clusters.

Clustering Analysis

- Hierarchical Clustering Algorithm
- Partitional Algorithm: k-means
- Mixture-Resolving and Mode-Seeking Algorithm
- Nearest Neighbor Clustering
- Fuzzy Clustering
- Artificial Neural Networks for Clustering
- Clustering Large data sets
- …

Data/Information Visualization

What is Visualization?



Visualization

[ˌvɪzjʊəlaɪˋzeɪʃən, -ʒʊə-; -lɪˋz-]

使看得見的，清楚地呈現在心

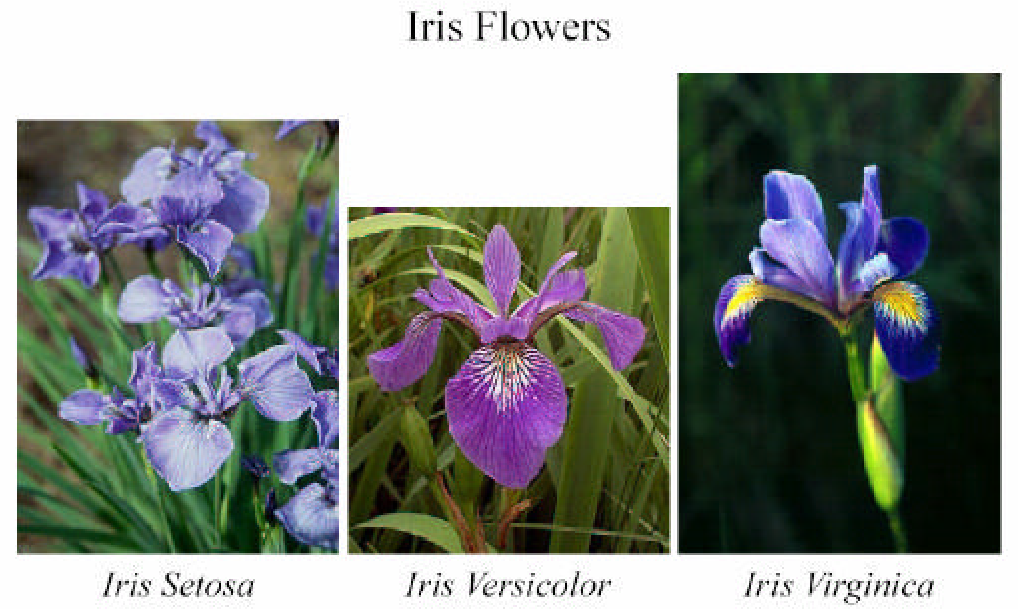
- To visualize = to make visible, to transform into pictures.
- Making things/processes visible that are not directly accessible by the human eye.
- Transformation of an abstraction to a picture.
- Computer aided extraction and display of information from data.

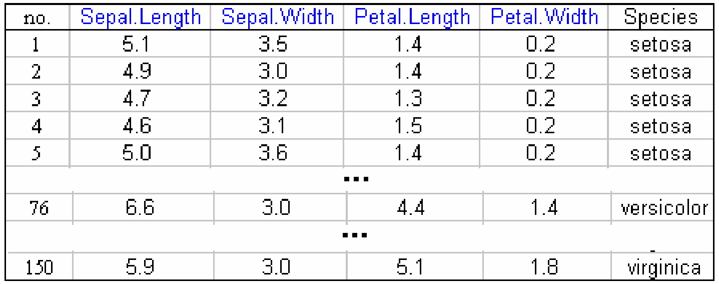
Data/Information Visualization

- Exploiting the human visual system to extract information from data.
- Provides an overview of complex data sets.
- Identifies structure, patterns, trends, anomalies, and relationships in data.
- Assists in identifying the areas of interest.

Tegarden, D. P. (1999). Business Information Visualization. Communications of AIS 1, 1-38.

The Iris Data (Anderson 1935; Fisher 1936)





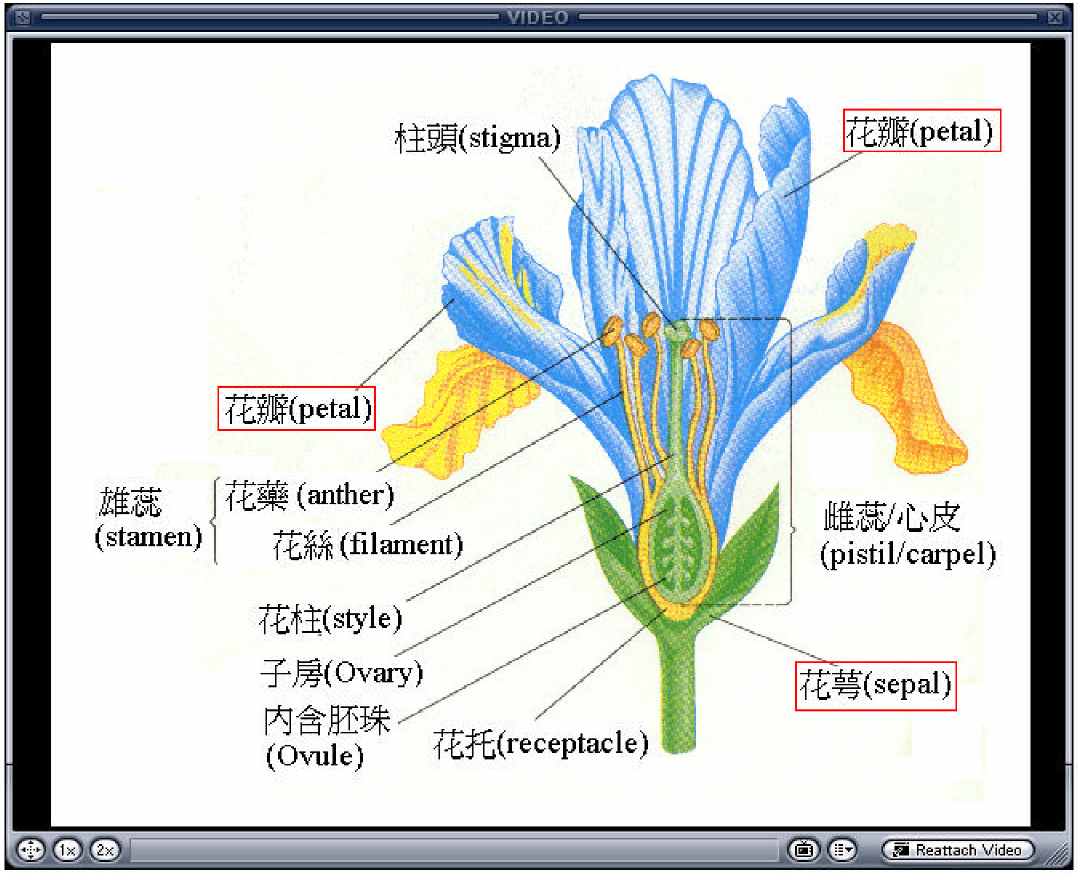
no.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
			...		
76	6.6	3.0	4.4	1.4	versicolor
			...		
150	5.9	3.0	5.1	1.8	virginica

Images source: http://www.stat.auckland.ac.nz/~ihaka/120/Lectures/lecture27.pdf

- The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis.
- The sepal length, sepal width, petal length, and petal width are measured in centimeters on fifty iris specimens from each of three species, *Iris setosa*, *I. versicolor*, and *I. virginica*.

花名	愛麗絲(IRIS)
花語	愛的留言，使者，易變的
種類	菖蒲科
原產	地中海沿岸 、中東
花色	白，黃，紫
花期	春　初夏
備註	法國國花

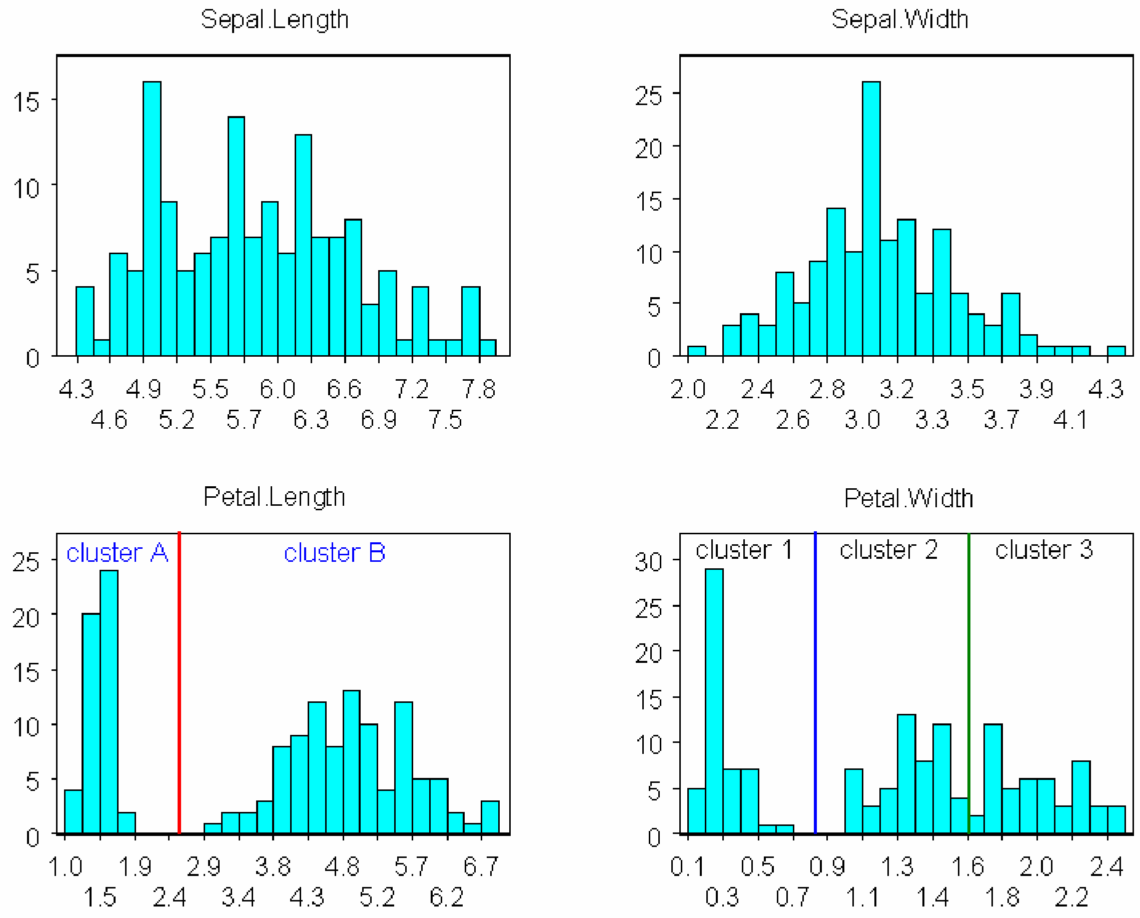
鳶尾花因為花瓣形如鳶鳥尾巴而稱之,其屬名Iris為希臘語 "彩虹"之意.喻指花色豐富.其俗稱亦以其屬名音譯為"愛麗絲",愛麗絲在希臘神話中是彩虹女神,她是眾神與凡間的使者,主要任務在於將善良人死后的靈魂,經由天地間的彩虹橋帶回天國.至今,希臘人常在墓地種植此花,就是希望人死后的靈魂能托付愛麗絲帶回天國,這也是花語"愛的使者"的由來.



Histograms

The histogram graphically shows:

1. center of the data (location)
2. spread of the data (scale)
3. skewness of the data
4. presence of outliers
5. presence of multiple modes in the data.



Two important properties of a clustering definition:
1. Most of data has been organized into non-overlapping clusters.
2. Each cluster has a within variance and one between variance for each of the other clusters. A good cluster should have a small within variance and large between variance.

Scatterplot

○ *Iris Virginica*

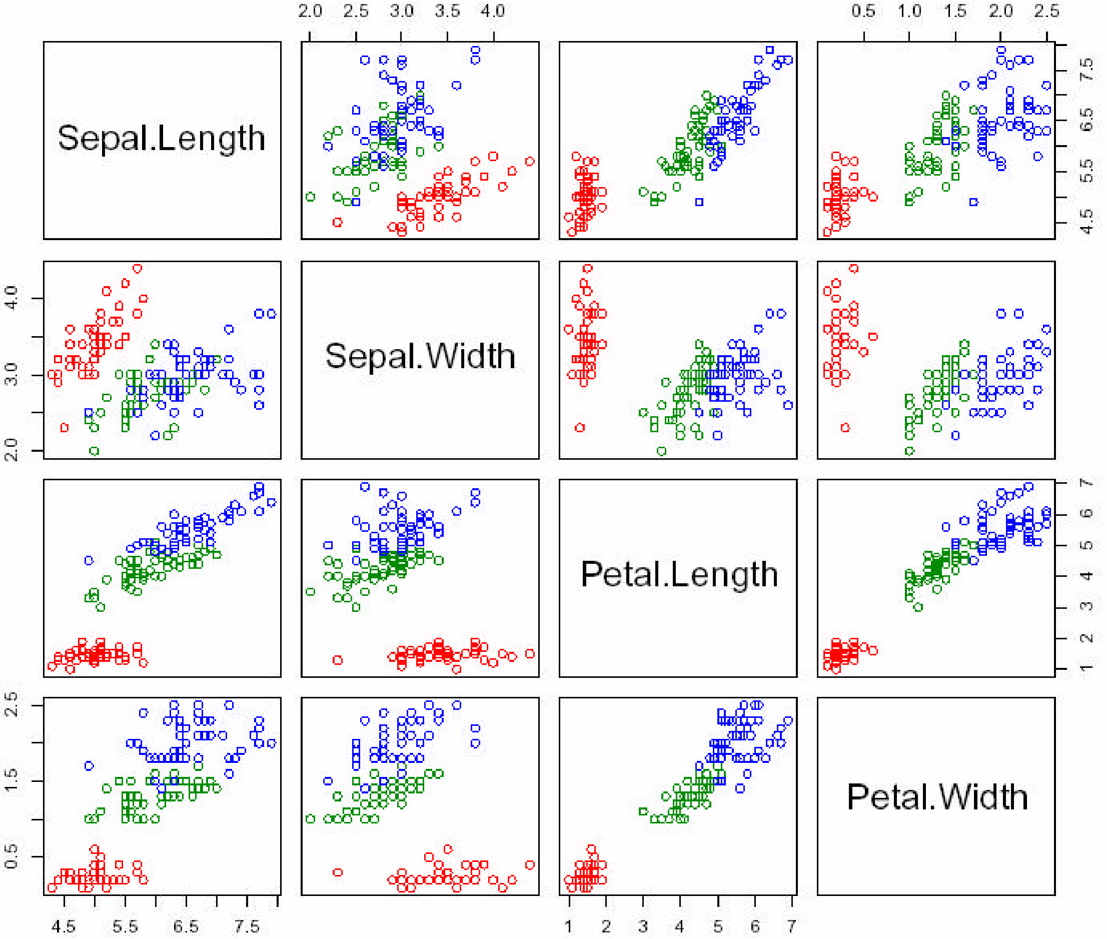
○ *Iris Versicolor*

○ *Iris Setosa*

The Iris Data

Sepal Length	Sepal Width	Petal Length	Petal Width
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2
⋮	⋮	⋮	⋮

Scatterplot Matrices



Images source: http://www.stat.auckland.ac.nz/~ihaka/120/Lectures/lecture27.pdf

Dendrogram (Kaufman and Rousseeuw, 1990)

- Hierarchical Clustering

Example: Agglomerative algorithm + Average linkage clustering

	a	b	c	d	e
a	0	2	6	10	9
b		0	5	9	8
c			0	4	5
d				0	3
e					0

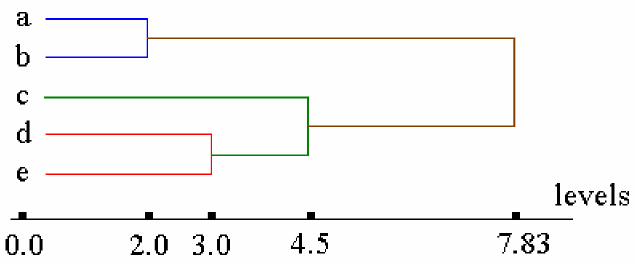
$$D(\{a,b\},\{c\}) = \frac{1}{2}[D(a,c)+D(b,c)] = \frac{1}{2}(6+5) = 5.5$$

	{a, b}	c	d	e
{a, b}	0	5.5	9.5	8.5
c		0	4	5
d			0	3
e				0

$$D(\{a,b\},\{d,e\}) = \frac{1}{4}[D(a,d)+D(a,e)+D(b,d)+D(b,e)]$$
$$= \frac{1}{4}(10+9+9+8) = 9$$

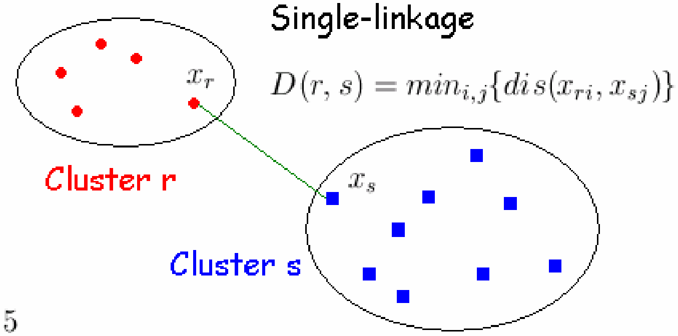
	{a,b}	c	{d, e}
{a, b}	0	5.5	9.0
c		0	4.5
{d, e}			0

	{a, b}	{c, d, e}
{a, b}	0	7.83
{c, d, e}		0



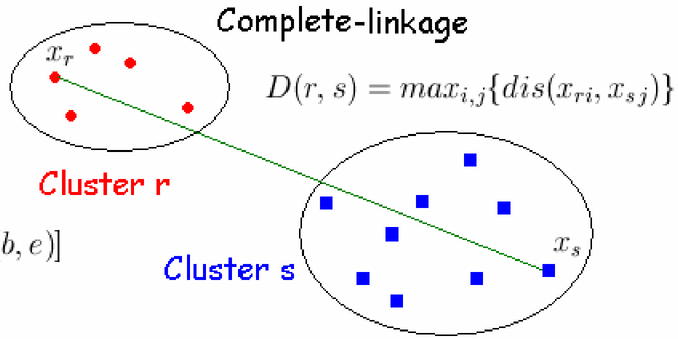
Single-linkage

$$D(r,s) = min_{i,j}\{dis(x_{ri}, x_{sj})\}$$



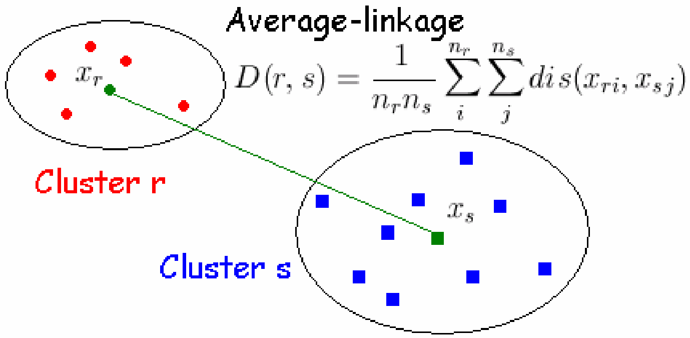
Complete-linkage

$$D(r,s) = max_{i,j}\{dis(x_{ri}, x_{sj})\}$$



Average-linkage

$$D(r,s) = \frac{1}{n_r n_s}\sum_{i}^{n_r}\sum_{j}^{n_s} dis(x_{ri}, x_{sj})$$



Visualizing and Clustering High-dimensional Data: dimension reduction techniques

- Principal Component Analysis (PCA)
- Multidimensional Scaling (MDS)
- Self-Organizing Maps (SOM)

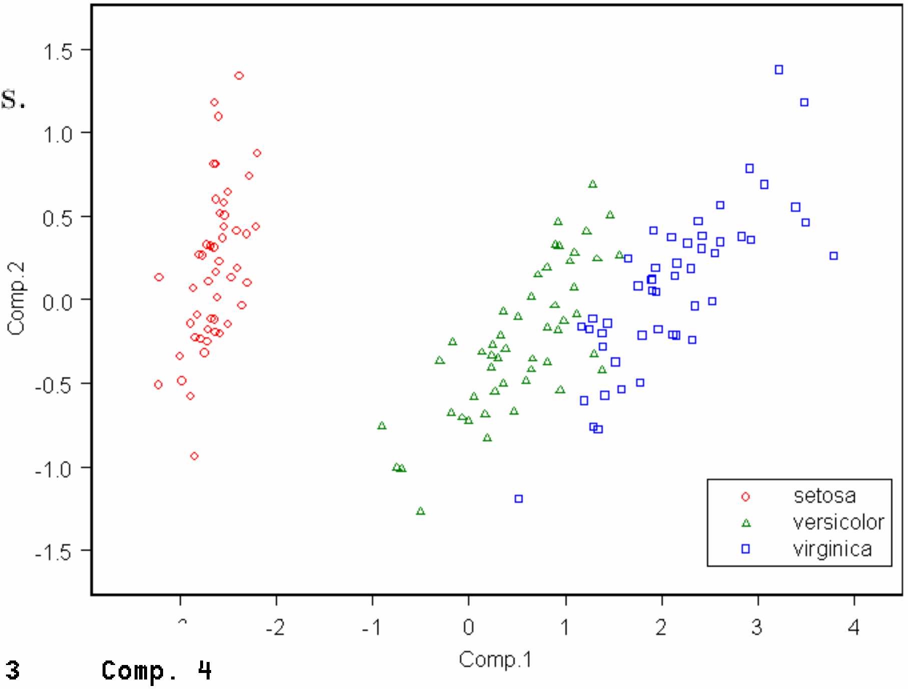
Dimension reduction visualization is often adopted for presenting grouping structure for methods such as K-means.

Principal Component Analysis (PCA)

(Pearson 1901; Hotelling 1933; Jolliffe 2002)

The ith principal component of $\mathbf{X}$ is $\mathbf{X}'\mathbf{v}_i$, where $\mathbf{v}_i$ is the ith normalized eigenvector of $\Sigma_{\mathbf{x}}$ corresponding to the ith largest eigenvaules.

The PCA summaries the dispersion of data points as data cloud in a small number of major axes (principal components) of varition among the variables.



Software: Splus

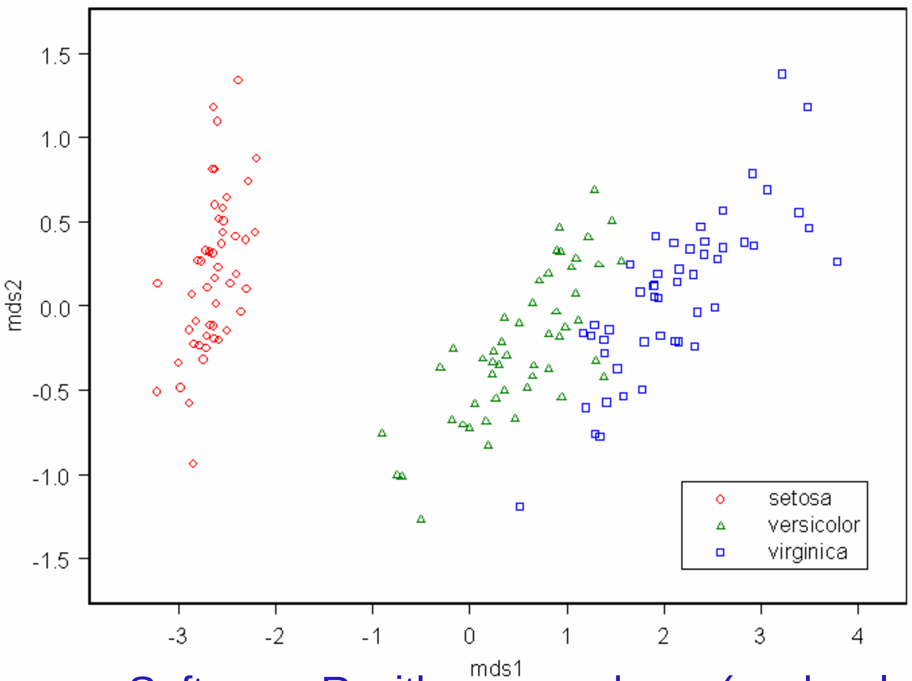
```
Importance of components:
                           Comp. 1    Comp. 2    Comp. 3     Comp. 4
Proportion of Variance 0.9246187 0.05306648 0.01710261 0.005212184
 Cumulative Proportion 0.9246187 0.97768521 0.99478782 1.000000000

Loadings:
             Comp.1  Comp.2  Comp.3  Comp.4
Sepal.Length  0.361   0.657  -0.582  -0.315
 Sepal.Width          0.730   0.598   0.320
Petal.Length  0.857  -0.173           0.480
 Petal.Width  0.358           0.546  -0.754
```

Multidimensional Scaling (MDS)

(Torgerson 1952; Cox and Cox 2001)

2D MDS configuration plot

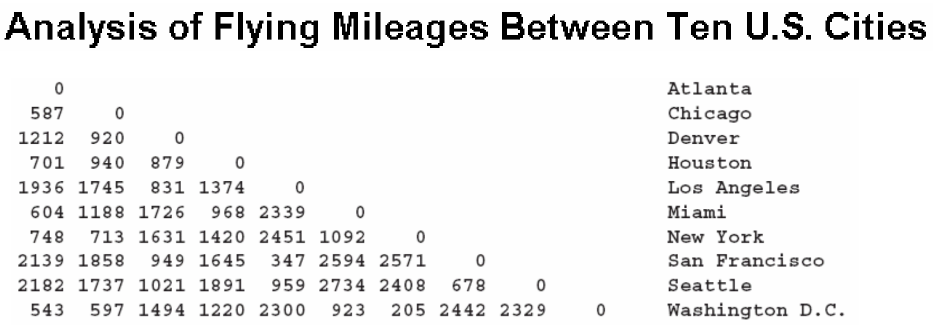


Classical MDS

Multidimensional scaling takes a set of dissimilarities and returns a set of points such that the distances between the points are approximately equal to the dissimilarities.

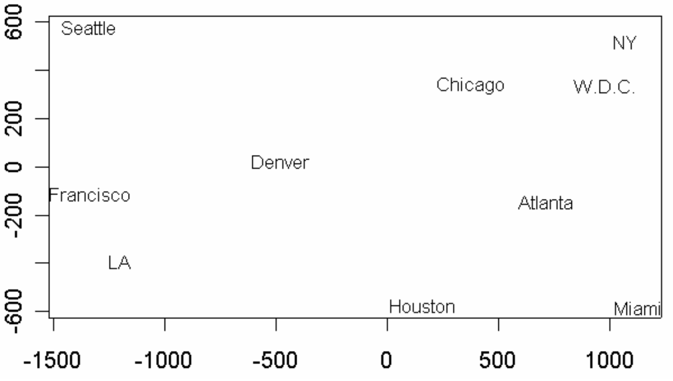
Note that if the input-space distances are Euclidean, classical MDS is equivalent to PCA. (Mardia et al. 1979)

Software: R with mva package (cmdscale)



Analysis of Flying Mileages Between Ten U.S. Cities

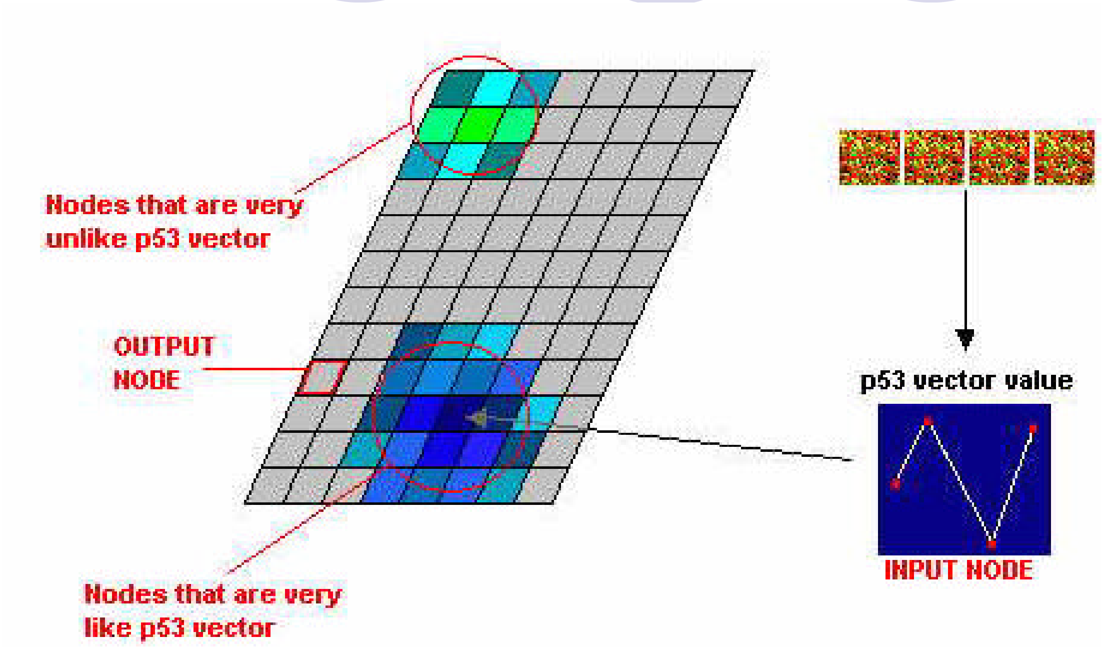
```
   0                                                      Atlanta
 587    0                                                 Chicago
1212  920    0                                            Denver
 701  940  879    0                                       Houston
1936 1745  831 1374    0                                  Los Angeles
 604 1188 1726  968 2339    0                             Miami
 748  713 1631 1420 2451 1092    0                        New York
2139 1858  949 1645  347 2594 2571    0                   San Francisco
2182 1737 1021 1891  959 2734 2408  678    0              Seattle
 543  597 1494 1220 2300  923  205 2442 2329    0         Washington D.C.
```



Self-Organizing Maps (SOM)

(kohonen 2001)

SOMs were developed by Kohonen in the early 1980's, original area was in the area of speech recognition.

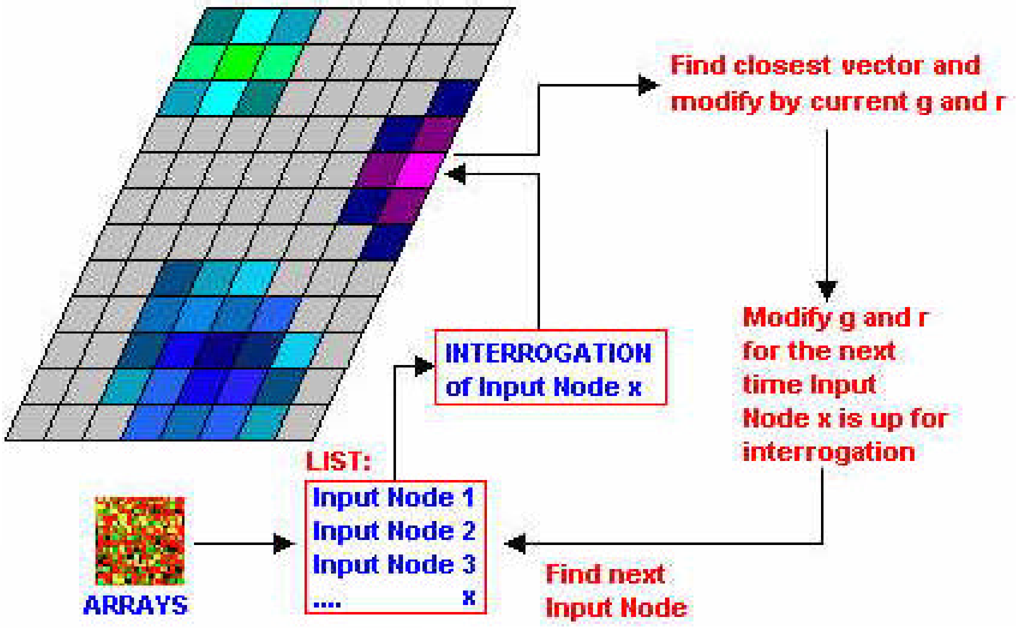

Nodes that are very unlike p53 vector
OUTPUT NODE
p53 vector value
INPUT NODE
Nodes that are very like p53 vector

SOM is unique in the sense that it combines both aspects. It can be used at the same time both to reduce the amount of data by clustering, and to construct a nonlinear projection of the data onto a low-dimensional display.

Figures source from: *SC*/path Home
http://www.ucl.ac.uk/oncology/MicroCore/tutorial.htm

(Organise data on the basis of similarity by putting entities geometrically close to each other)

Overview of SOM



Find closest vector and modify by current g and r

Modify g and r for the next time Input Node x is up for interrogation

INTERROGATION of Input Node x

LIST:

Input Node 1
Input Node 2
Input Node 3
.... x

ARRAYS

Find next Input Node

Figures source from: *SCI*path Home
http://www.ucl.ac.uk/oncology/MicroCore/tutorial.htm

Step 0: Initialize weights $\mathbf{w}_i(t)$.

Set topological neighborhood parameters $N_c(t)$.

Set learning rate parameters $\alpha(t)$ and $h_{ci}(t)$.

Step 1: For each input vector $\mathbf{x}(t)$, do

a. Finding a BMU: $\|\mathbf{x}(t) - \mathbf{w}_c(t)\| = \min_i \|\mathbf{x}(t) - \mathbf{w}_i(t)\|$

b. Learning process:

$$\mathbf{w}_i(t+1) = \begin{cases} \mathbf{w}_i(t) + h_{ci}(t)[\,\mathbf{x}(t) - \mathbf{w}_i(t)\,], & i \in N_c(t) \\ \mathbf{w}_i(t), & \text{o.w.} \end{cases}$$

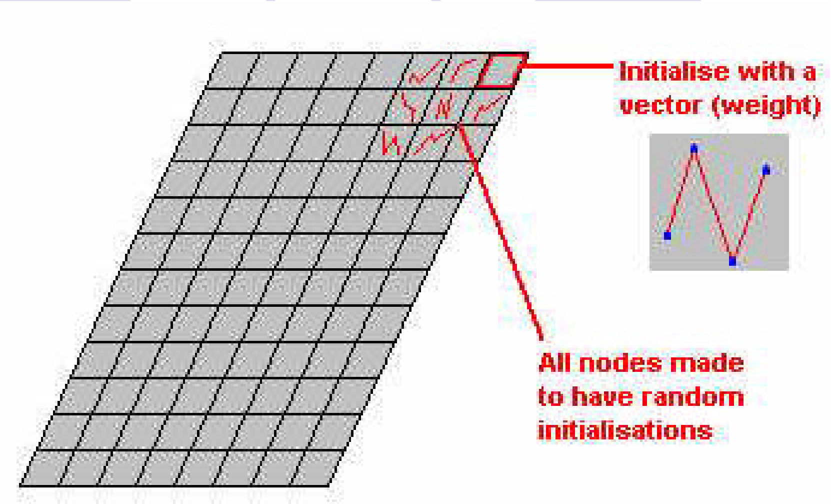
c. Go to the next unvisited input vector. If there are no unvisited input vector left then go back to the very first one and go to Step 2.

Step 2: Incrementally decrease the learning rate and the neighborhood size, and repeat Step 1.

Step 3: Keep doing Steps 1 and 2 for a sufficient number of iterations.

SOM - Initialization

Step 0: Initialize weights $\mathbf{w}_i(t)$.
Set topological neighborhood parameters $N_c(t)$.
Set learning rate parameters $\alpha(t)$ and $h_{ci}(t)$.

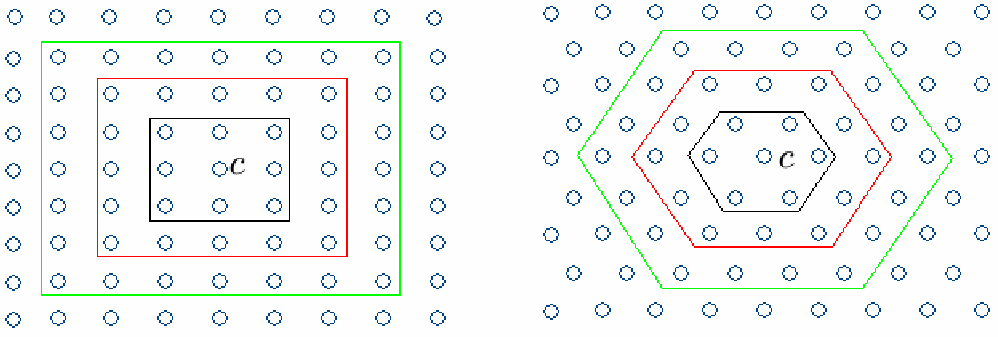


Initialise with a vector (weight)

All nodes made to have random initialisations

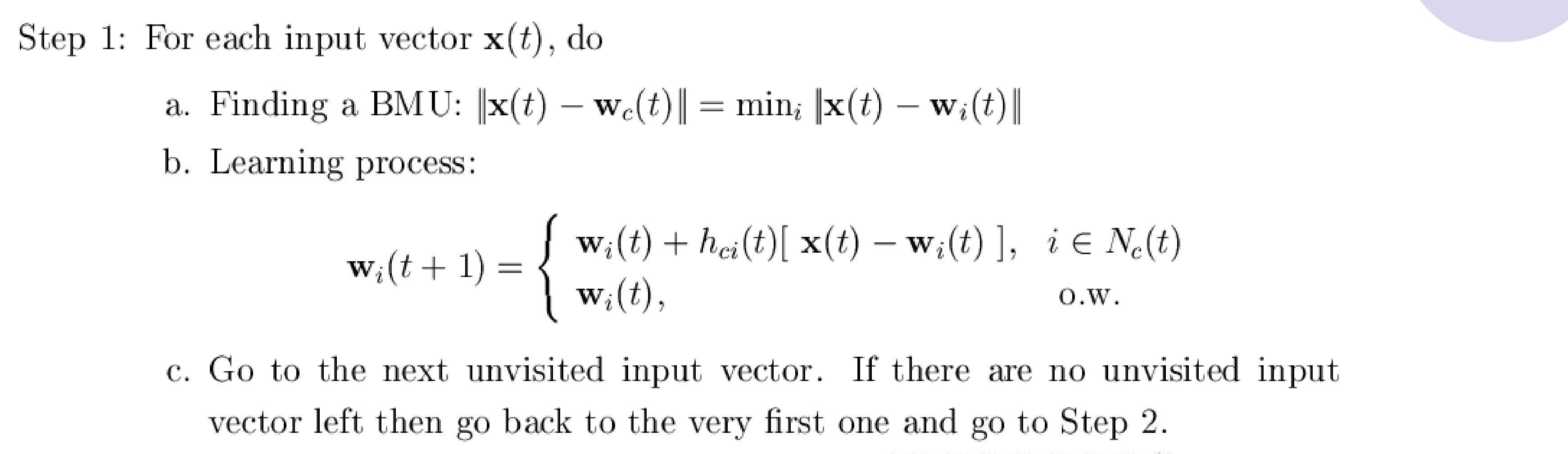
SOM initialization means to give each weight of the output node a random (or determined) vector value. *The dimensionality of the vector values put in **must match** the dimensionality of the raw data!* So if the raw data consists of 5 arrays, then the vectors must have 5 elements (dimensions).

Two examples of topological neighborhood.



■ $N_c(t_1) = 1$, 🟥 $N_c(t_2) = 2$, 🟩 $N_c(t_3) = 3$, $t_1 < t_2 < t_3$

SOM Algorithm



Step 1: For each input vector $\mathbf{x}(t)$, do

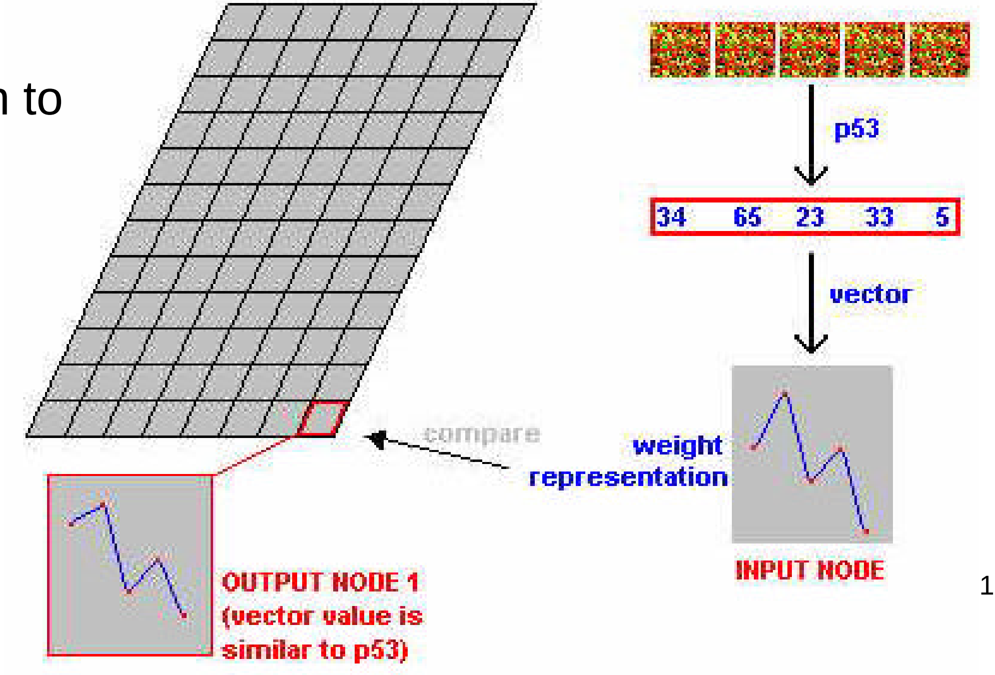
a. Finding a BMU: $\|\mathbf{x}(t) - \mathbf{w}_c(t)\| = \min_i \|\mathbf{x}(t) - \mathbf{w}_i(t)\|$

b. Learning process:

$$\mathbf{w}_i(t+1) = \begin{cases} \mathbf{w}_i(t) + h_{ci}(t)[\ \mathbf{x}(t) - \mathbf{w}_i(t)\ ], & i \in N_c(t) \\ \mathbf{w}_i(t), & \text{o.w.} \end{cases}$$

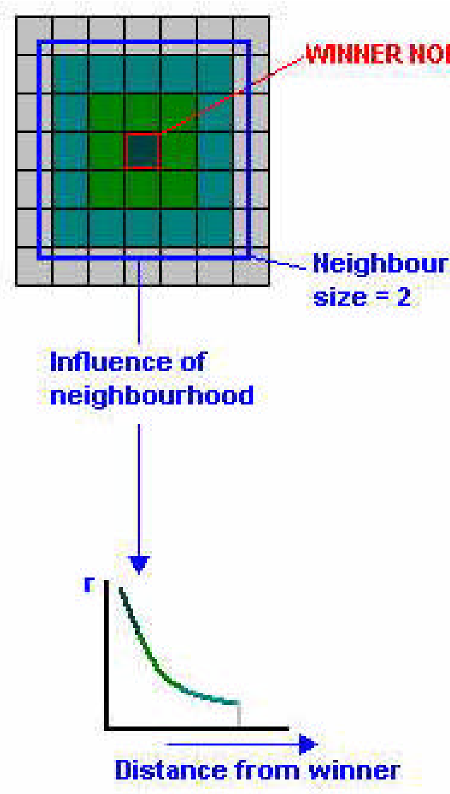
c. Go to the next unvisited input vector. If there are no unvisited input vector left then go back to the very first one and go to Step 2.

The SOM algorithm then goes on to interrogate the map for similar vectors



Figures source from: *SCI*path Home
http://www.ucl.ac.uk/oncology/MicroCore/tutorial.htm

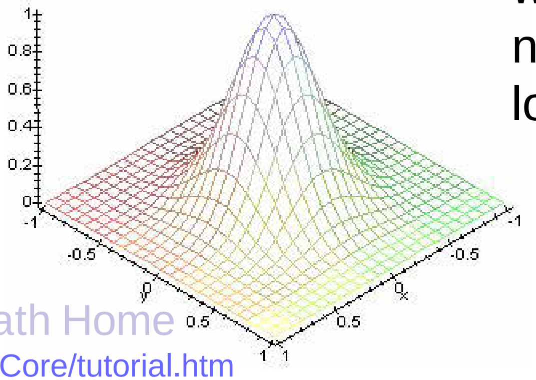
SOM Algorithm: neighborhood functions



The winner node's weight is *modified* such that it becomes even more *similar* to the original input node's vector.

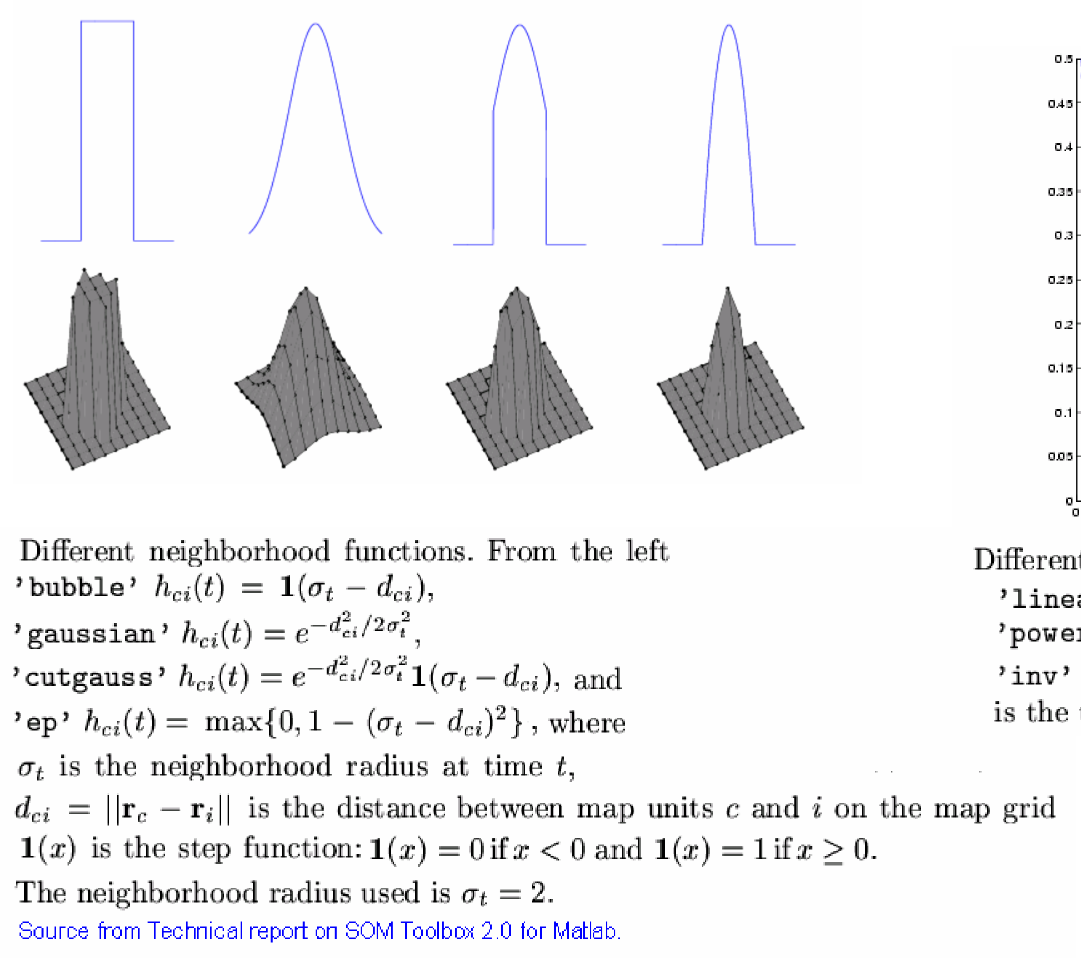
The neighborhood value has a two-fold character - a *size* and a *function of distance to influence*. One could even define a further third character - the *shape* of the neighborhood (in this case, a square - highlighted in blue).

The peak of the Gaussian function would be the location of the winner node. As one moves out from that location, the *r* value decreases.



Figures source from: *SCI*path Home
http://www.ucl.ac.uk/oncology/MicroCore/tutorial.htm

SOM Algorithm: neighborhood functions and learning rate functions



Different neighborhood functions. From the left `'bubble'` $h_{ci}(t) = \mathbf{1}(\sigma_t - d_{ci})$, `'gaussian'` $h_{ci}(t) = e^{-d_{ci}^2/2\sigma_t^2}$, `'cutgauss'` $h_{ci}(t) = e^{-d_{ci}^2/2\sigma_t^2}\mathbf{1}(\sigma_t - d_{ci})$, and `'ep'` $h_{ci}(t) = \max\{0, 1 - (\sigma_t - d_{ci})^2\}$, where

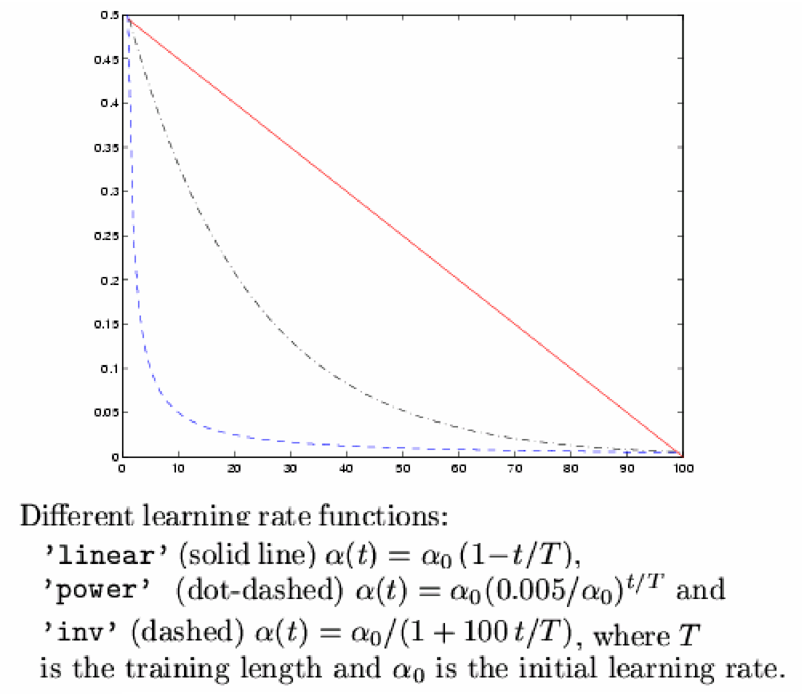
σ_t is the neighborhood radius at time t,

$d_{ci} = ||\mathbf{r}_c - \mathbf{r}_i||$ is the distance between map units c and i on the map grid

$\mathbf{1}(x)$ is the step function: $\mathbf{1}(x) = 0$ if $x < 0$ and $\mathbf{1}(x) = 1$ if $x \geq 0$.

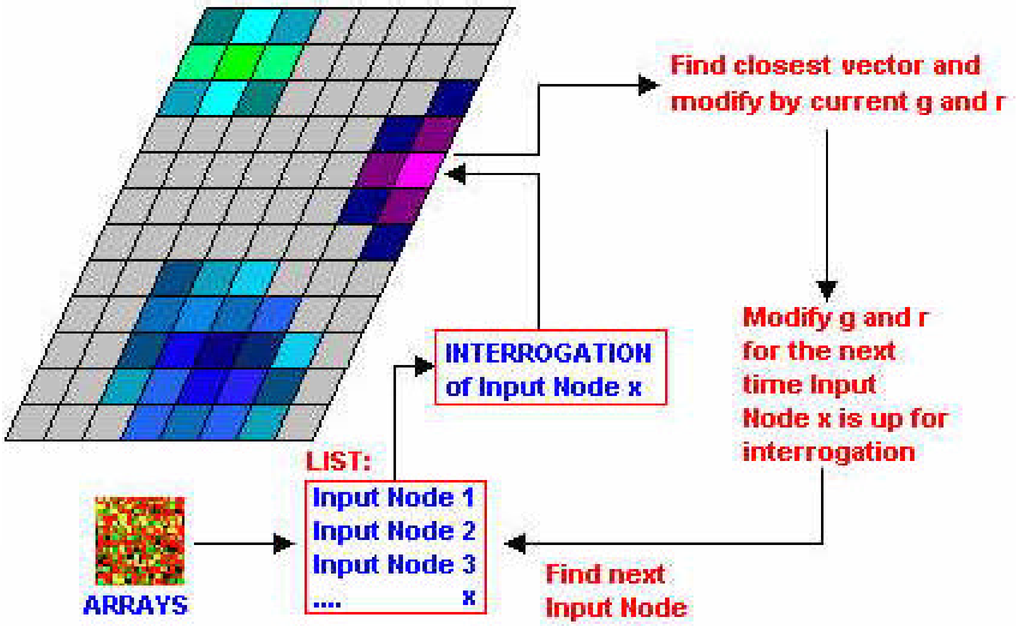
The neighborhood radius used is $\sigma_t = 2$.

Source from Technical report on SOM Toolbox 2.0 for Matlab.



Different learning rate functions: `'linear'` (solid line) $\alpha(t) = \alpha_0\,(1 - t/T)$, `'power'` (dot-dashed) $\alpha(t) = \alpha_0\,(0.005/\alpha_0)^{t/T}$ and `'inv'` (dashed) $\alpha(t) = \alpha_0/(1 + 100\,t/T)$, where T is the training length and α_0 is the initial learning rate.

Summary of SOM


Find closest vector and modify by current g and r
Modify g and r for the next time Input Node x is up for interrogation
INTERROGATION of Input Node x
LIST:
Input Node 1
Input Node 2
Input Node 3
.... x
ARRAYS
Find next Input Node

Figures source from: *SCI*path Home
http://www.ucl.ac.uk/oncology/MicroCore/tutorial.htm

Step 0: Initialize weights $\mathbf{w}_i(t)$.

Set topological neighborhood parameters $N_c(t)$.

Set learning rate parameters $\alpha(t)$ and $h_{ci}(t)$.

Step 1: For each input vector $\mathbf{x}(t)$, do

a. Finding a BMU: $\|\mathbf{x}(t) - \mathbf{w}_c(t)\| = \min_i \|\mathbf{x}(t) - \mathbf{w}_i(t)\|$

b. Learning process:

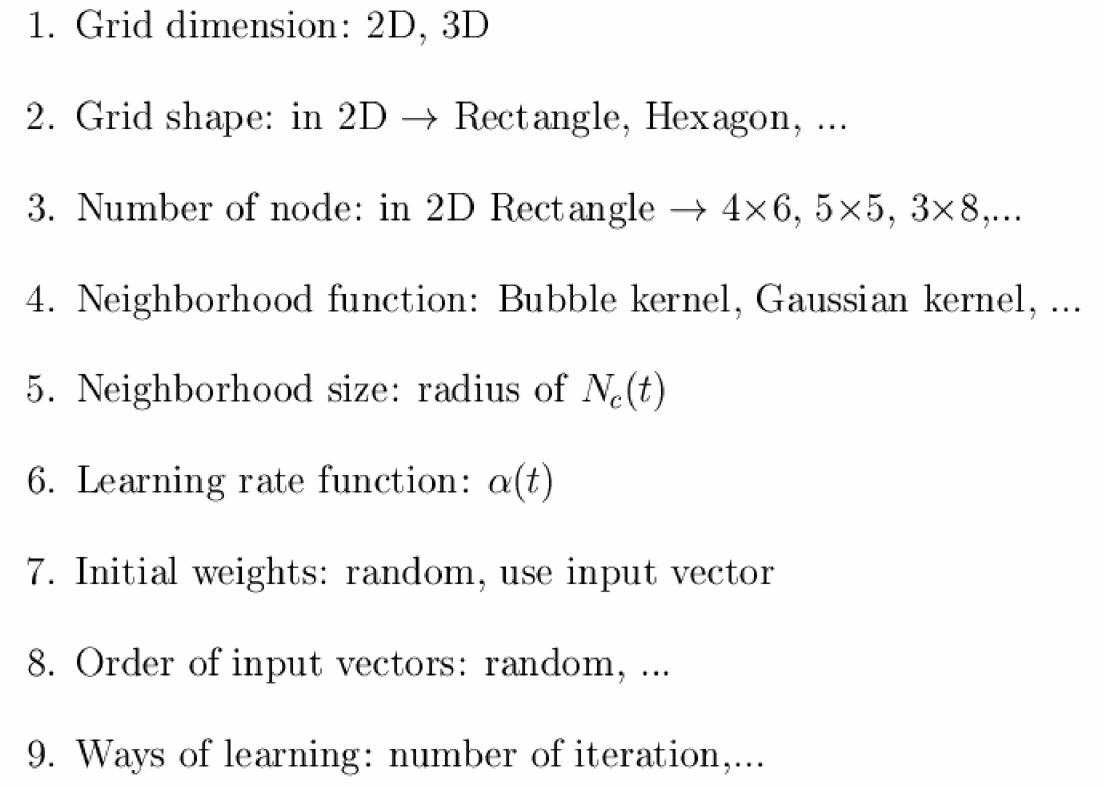
$$\mathbf{w}_i(t+1) = \begin{cases} \mathbf{w}_i(t) + h_{ci}(t)[\,\mathbf{x}(t) - \mathbf{w}_i(t)\,], & i \in N_c(t) \\ \mathbf{w}_i(t), & \text{o.w.} \end{cases}$$

c. Go to the next unvisited input vector. If there are no unvisited input vector left then go back to the very first one and go to Step 2.

Step 2: Incrementally decrease the learning rate and the neighborhood size, and repeat Step 1.

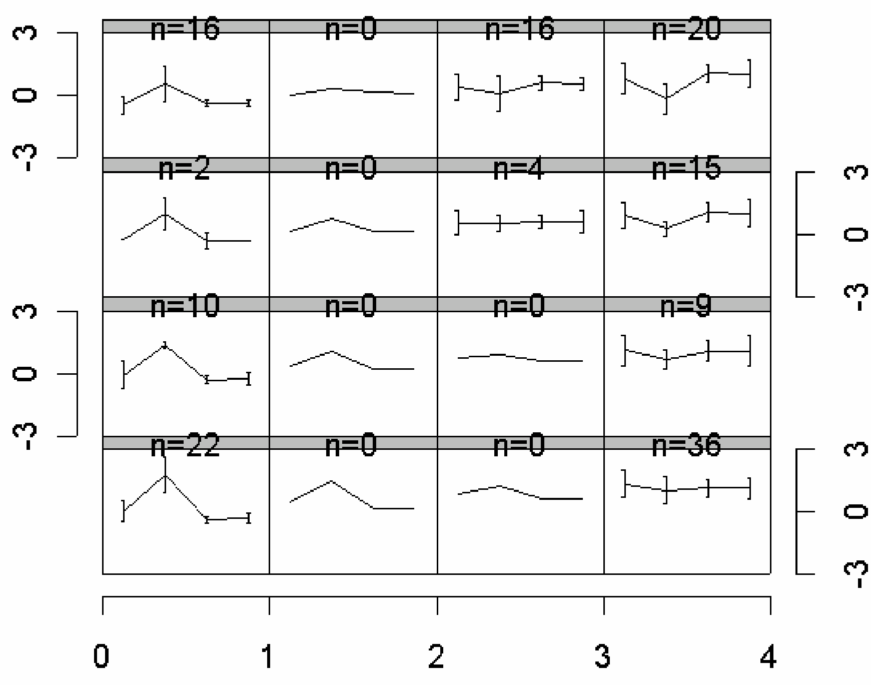
Step 3: Keep doing Steps 1 and 2 for a sufficient number of iterations.

Possible parameters used in SOM analysis



1. Grid dimension: 2D, 3D
2. Grid shape: in 2D → Rectangle, Hexagon, ...
3. Number of node: in 2D Rectangle → 4×6, 5×5, 3×8,...
4. Neighborhood function: Bubble kernel, Gaussian kernel, ...
5. Neighborhood size: radius of $N_c(t)$
6. Learning rate function: $\alpha(t)$
7. Initial weights: random, use input vector
8. Order of input vectors: random, ...
9. Ways of learning: number of iteration,...

SOM: iris example



Software: R: The som Package

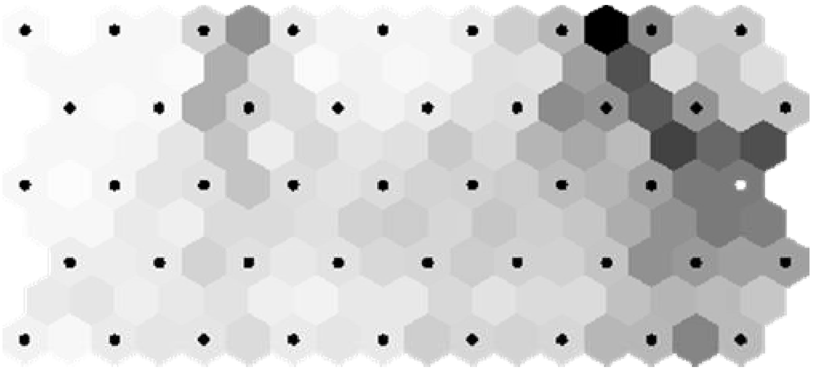
http://cran.r-project.org/src/contrib/som_0.2-7.tar.gz

```
> iris.data.n <- normalize(iris.data, byrow=F)
> iris.som <- som(iris.data.n, xdim=4, ydim=4, topol="rect", neigh="gaussian")
> plot.som(iris.som)
```

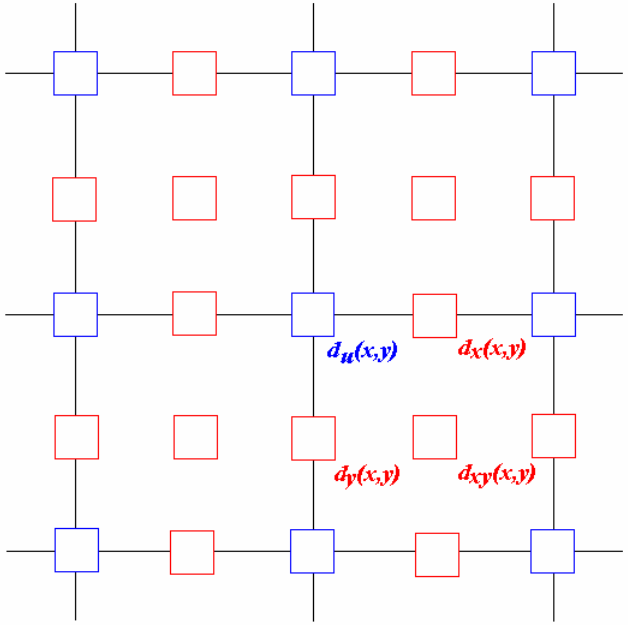
U-matrix: Unified Matrix Method

(Ultsch and Siemon 1989, Ultsch 1993)

U-matrix representation of SOM visualizes the distance between the neurons. The distance between the adjacent neurons is calculated and presented with different colorings between the adjacent nodes.



U-matrix representation of the SOM



$b(x,y)$: matrix of neurons, of size $n_x \times n_y$.

$w_i(x,y)$: matrix of weights.

$u(x,y)$: U-matrix of size $(2n_x - 1) \times (2n_y - 1)$.

$d_x(x,y)$: $\|b(x,y) - b(x+1,y)\| = \sqrt{\sum_i [w_i(x,y) - w_i(x+1,y)]^2}$

$d_y(x,y)$: $\|b(x,y) - b(x,y+1)\| = \sqrt{\sum_i [w_i(x,y) - w_i(x,y+1)]^2}$

$d_{xy}(x,y)$: $\frac{1}{2}[\frac{\|b(x,y)-b(x+1,y+1)\|}{\sqrt{2}} + \frac{\|b(x,y+1)-b(x+1,y)\|}{\sqrt{2}}]$

$d_u(x,y)$: the median of the surrounding elements.

SOM: iris example

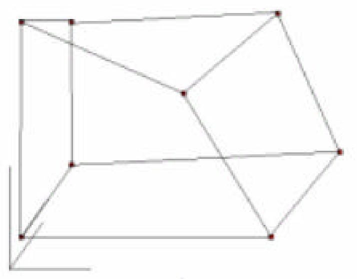


Figure 3.1: Example dataset

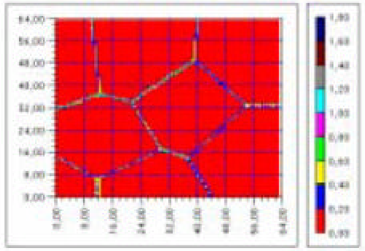
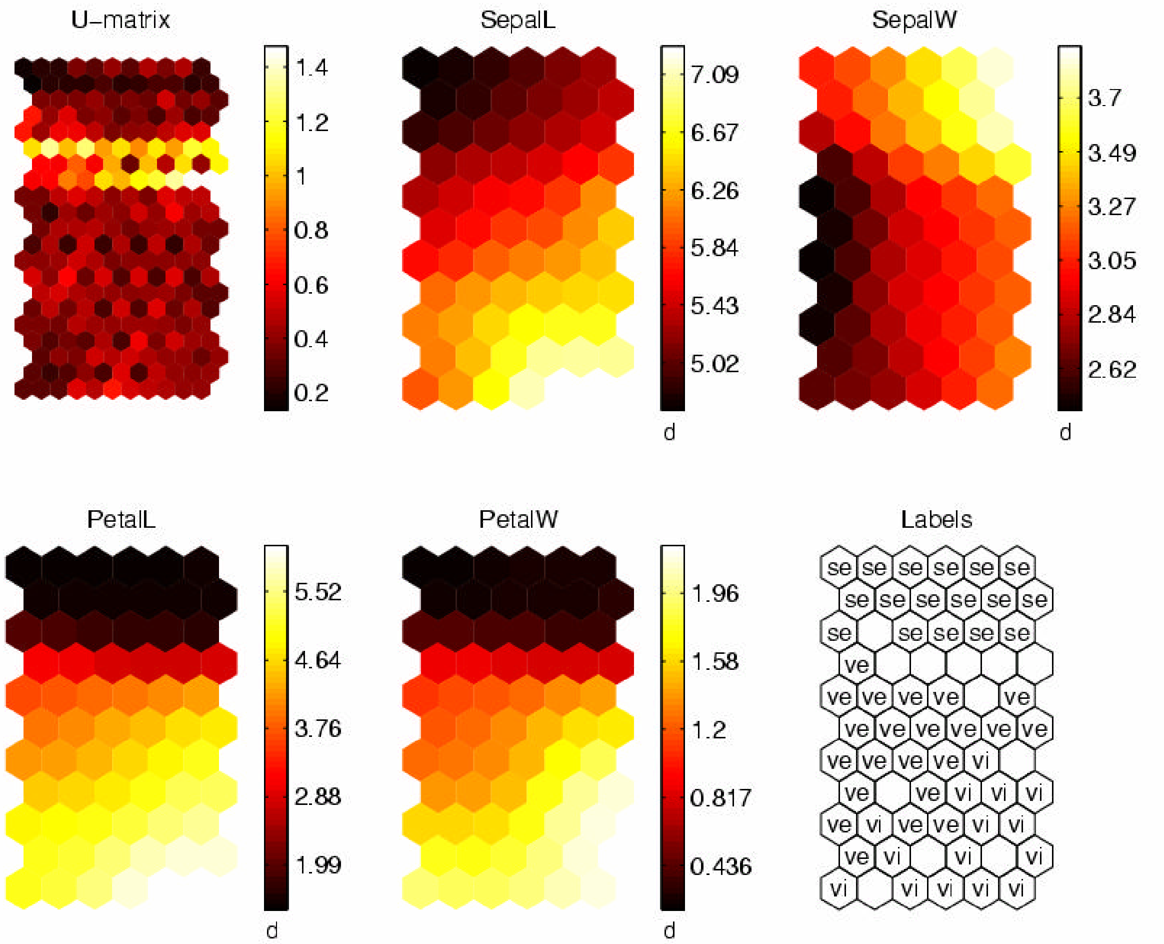


Figure 3.2: U-Matrix

A.Ultsch, C.Vetter (1994)
Self-Organizing-Feature-Maps versus Statistical Clustering Methods: A Benchmark



Software: SOM Toolbox 2.0 for Matlab

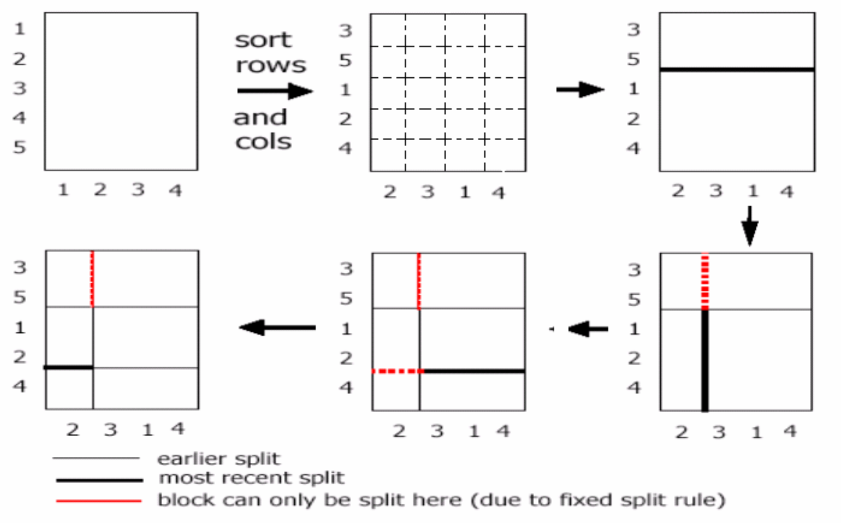
Source from technical Report on SOM Toolbox 2.0 for Matlab

Visualizing and Clustering High-dimensional Data: dimension-free visualization

- Block Clustering
- Data Image
- Generalized Association Plots (GAP)

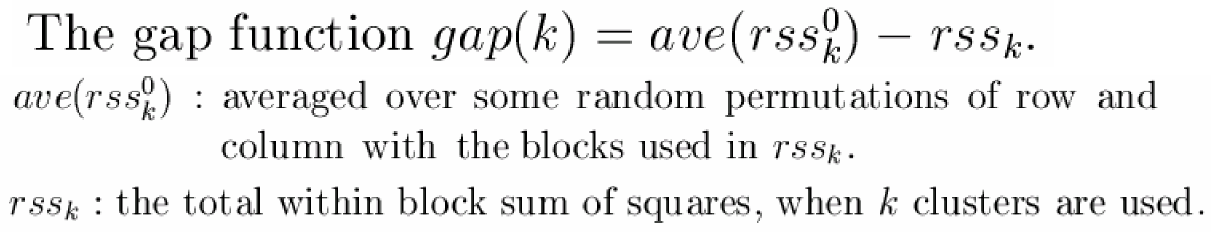
Block clustering (Hartigan, 1972)

- Reorders rows and columns to produce a matrix with homogenous blocks.
- Algorithm:
  - rows (columns) are sorted by row (column) mean.
  - Start with entire data in one block.



  - Choose the row or column split (of all existing blocks) that reduces total within- block- variance the most
  - Continue until a large number of blocks are obtained.
  - Recombine blocks by pruning.

Stopping rule: The “maximum gap” approach.



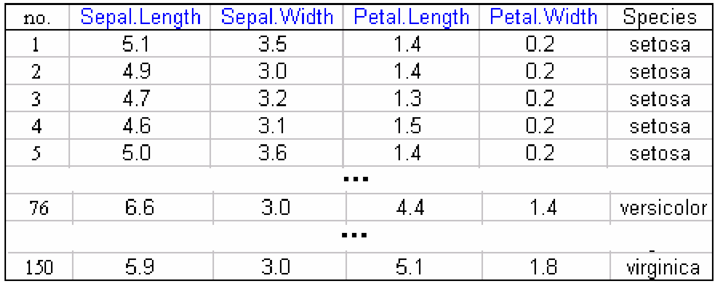
The gap function $gap(k) = ave(rss_k^0) - rss_k$.

$ave(rss_k^0)$: averaged over some random permutations of row and column with the blocks used in rss_k.

rss_k : the total within block sum of squares, when k clusters are used.

Data Image

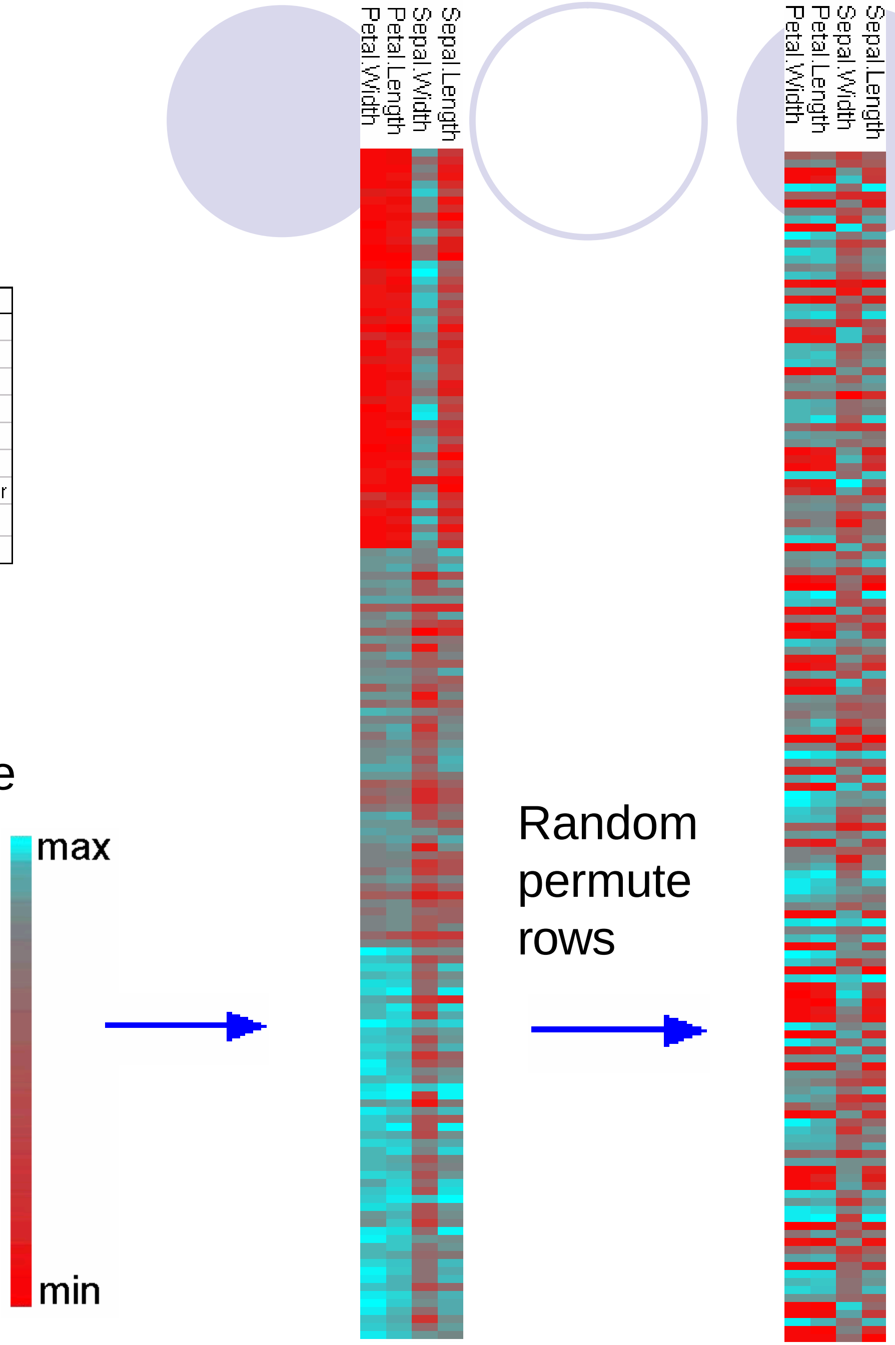
(Minnotte and Webster 1999)



no.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
			…		
76	6.6	3.0	4.4	1.4	versicolor
			…		
150	5.9	3.0	5.1	1.8	virginica

Column condition:

For each variable



Random permute rows

Splus code from Michael C. Minnotte:
http://math.usu.edu/~minnotte/research/pubs.html

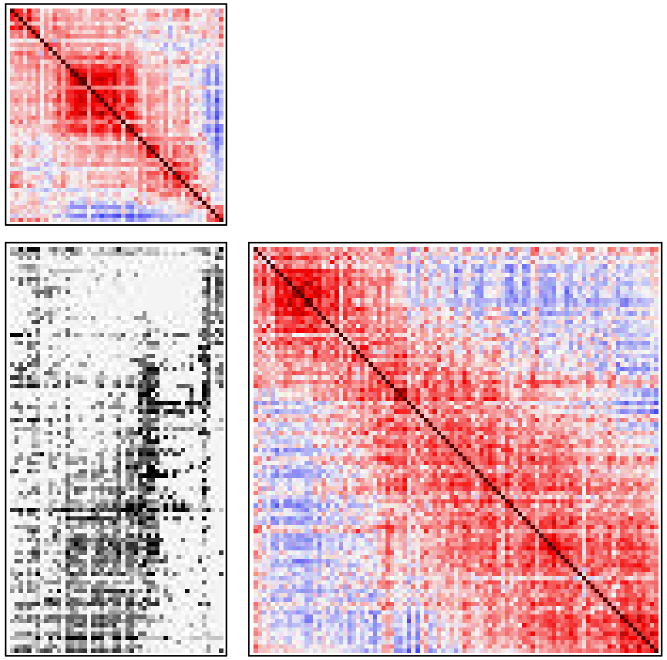
Relativity of a Statistical Graph

(Chang et al. 2002)

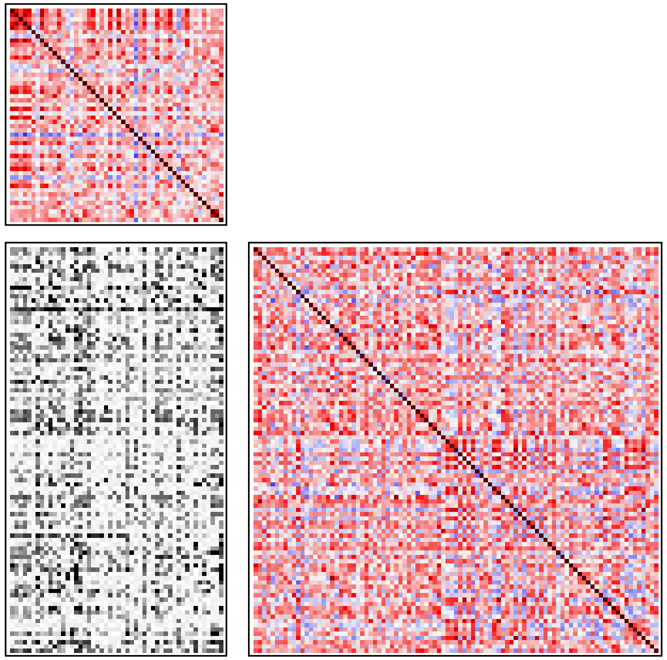
Concept:

placing similar (different) objects at closer (distant) positions

Permuted with ***good*** Orders



Permuted with ***bad*** Orders

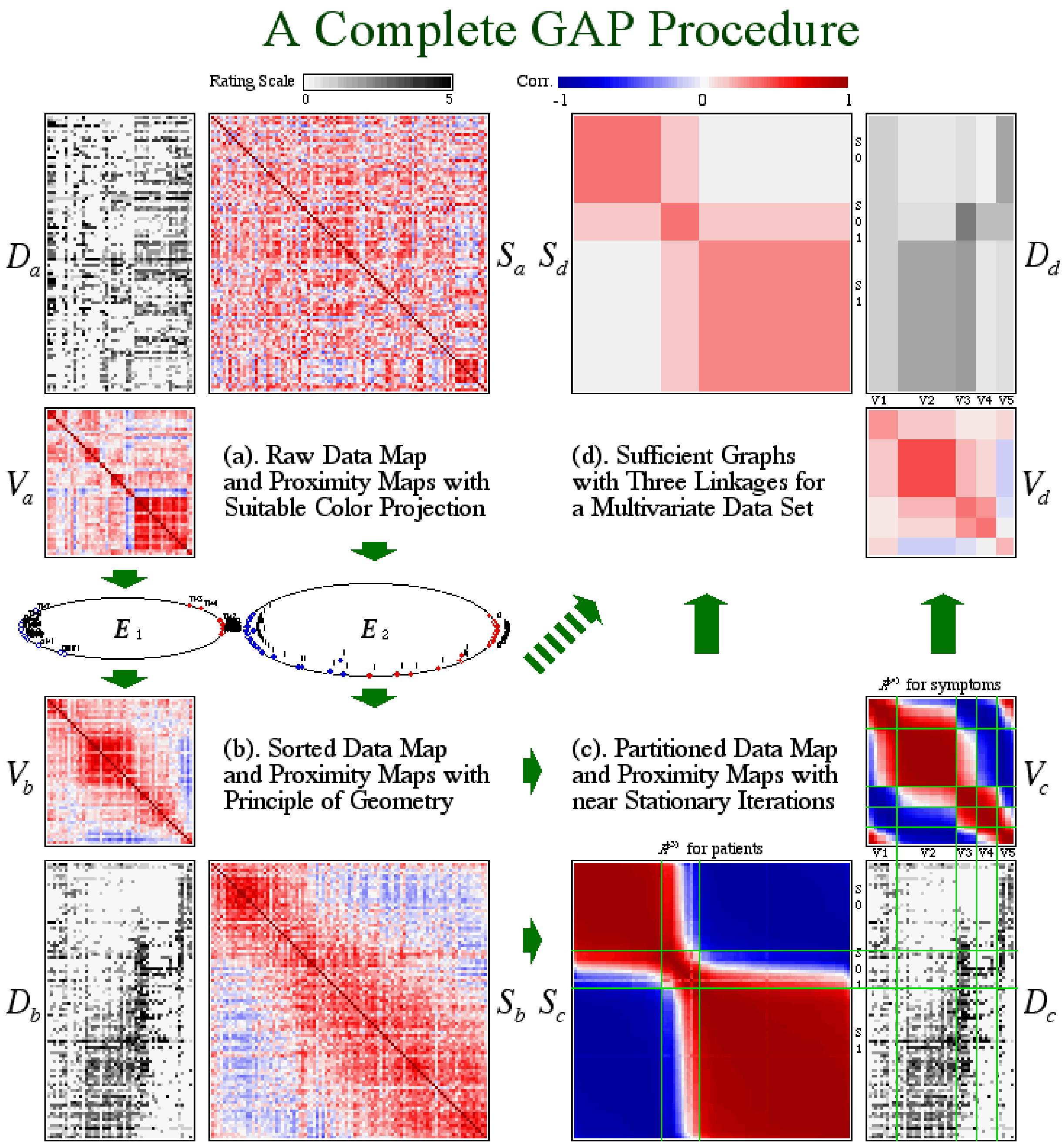


GAP

Generalized Association Plots

(Chen, 2002)

全矩陣式視覺化



A Complete GAP Procedure

(a). Raw Data Map and Proximity Maps with Suitable Color Projection

(b). Sorted Data Map and Proximity Maps with Principle of Geometry

(c). Partitioned Data Map and Proximity Maps with near Stationary Iterations

(d). Sufficient Graphs with Three Linkages for a Multivariate Data Set

GAP: iris example

Species Ordering

原始資料之呈現

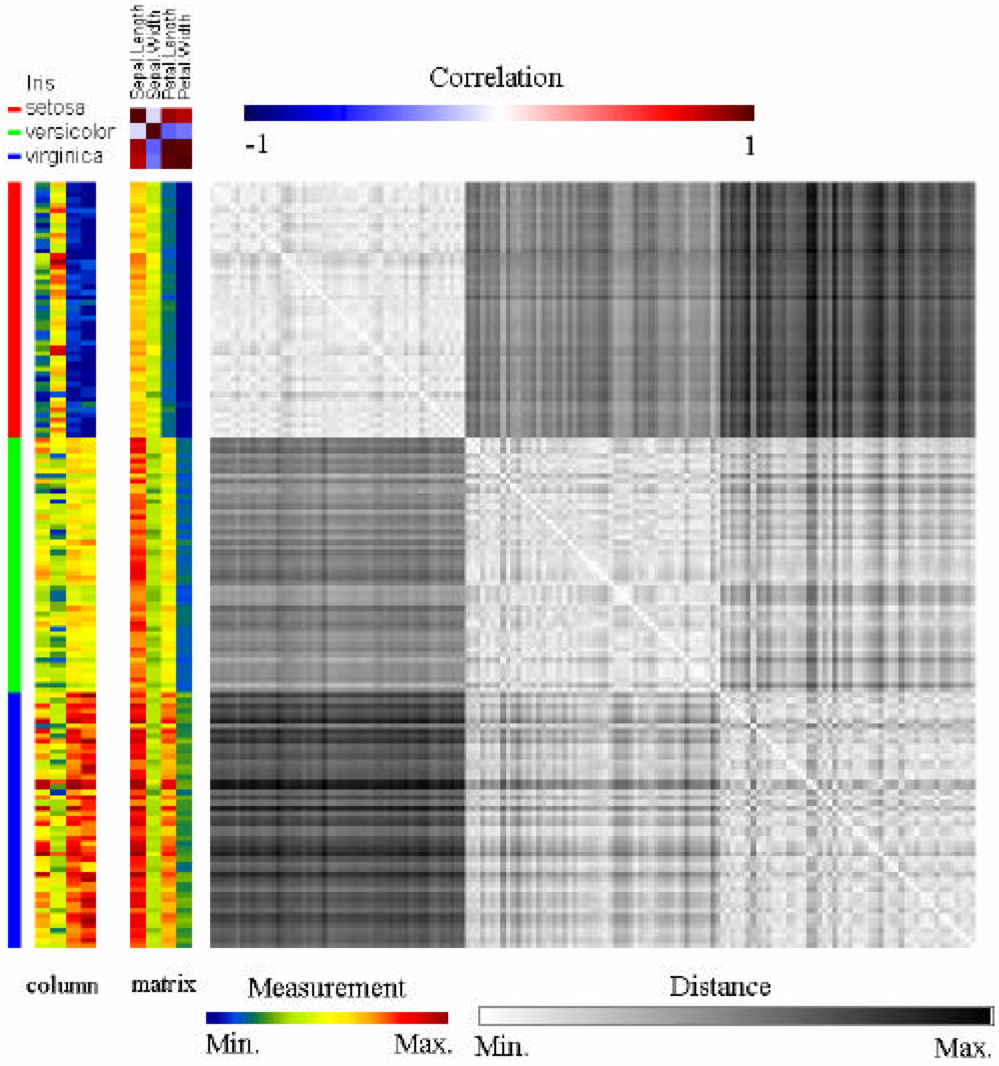
- color spectrum
- variable transformation
- similarity measure

Software

(1) Xlispstat

(2) C++ codes & **小畫家**

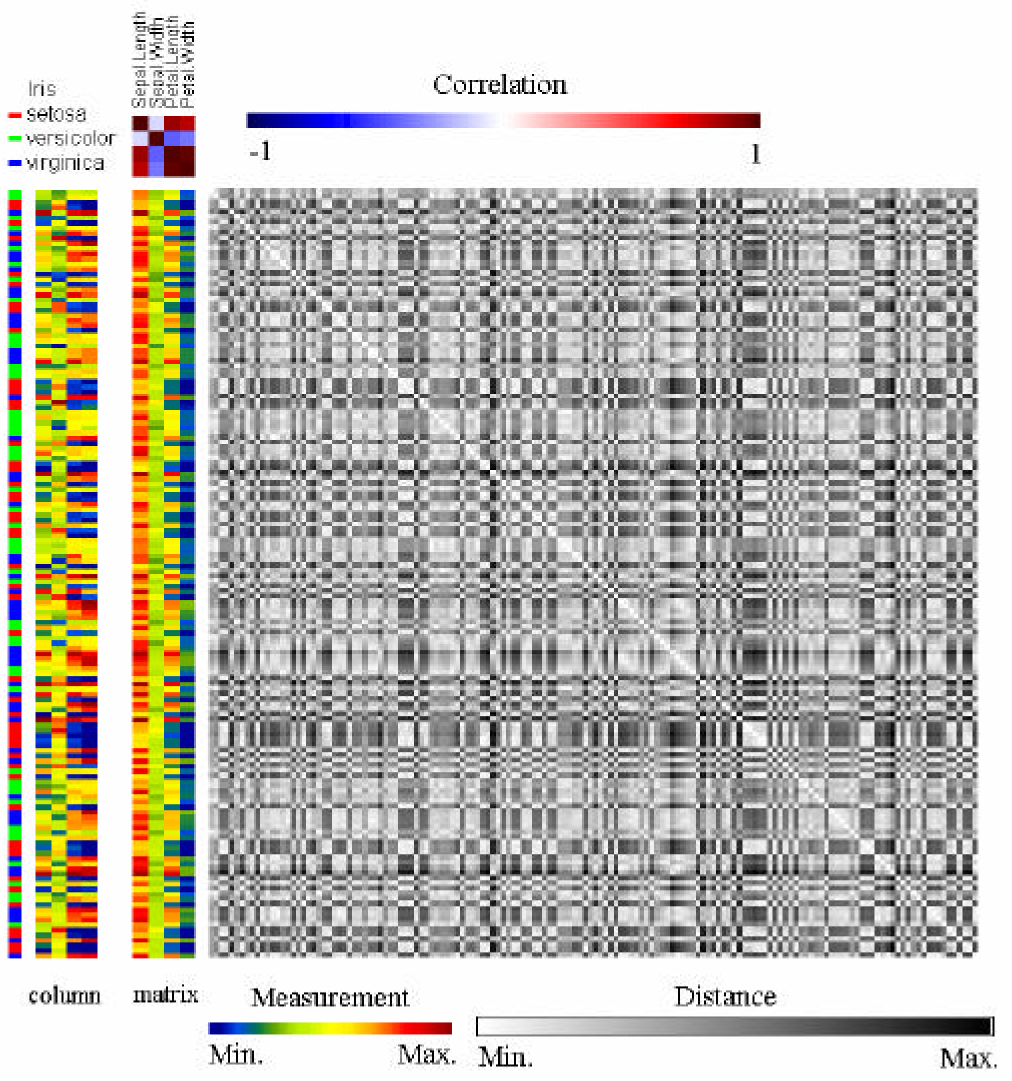
(3) GapLite (to appear)

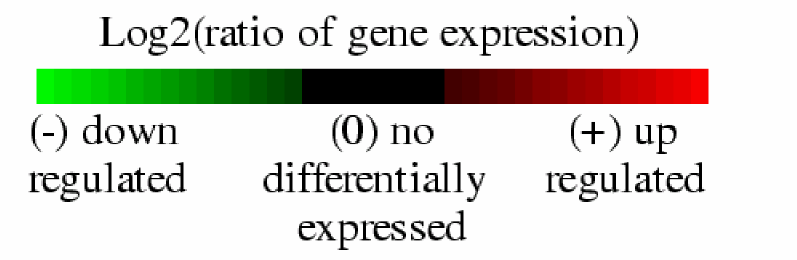


GAP: iris example

Random Permutation

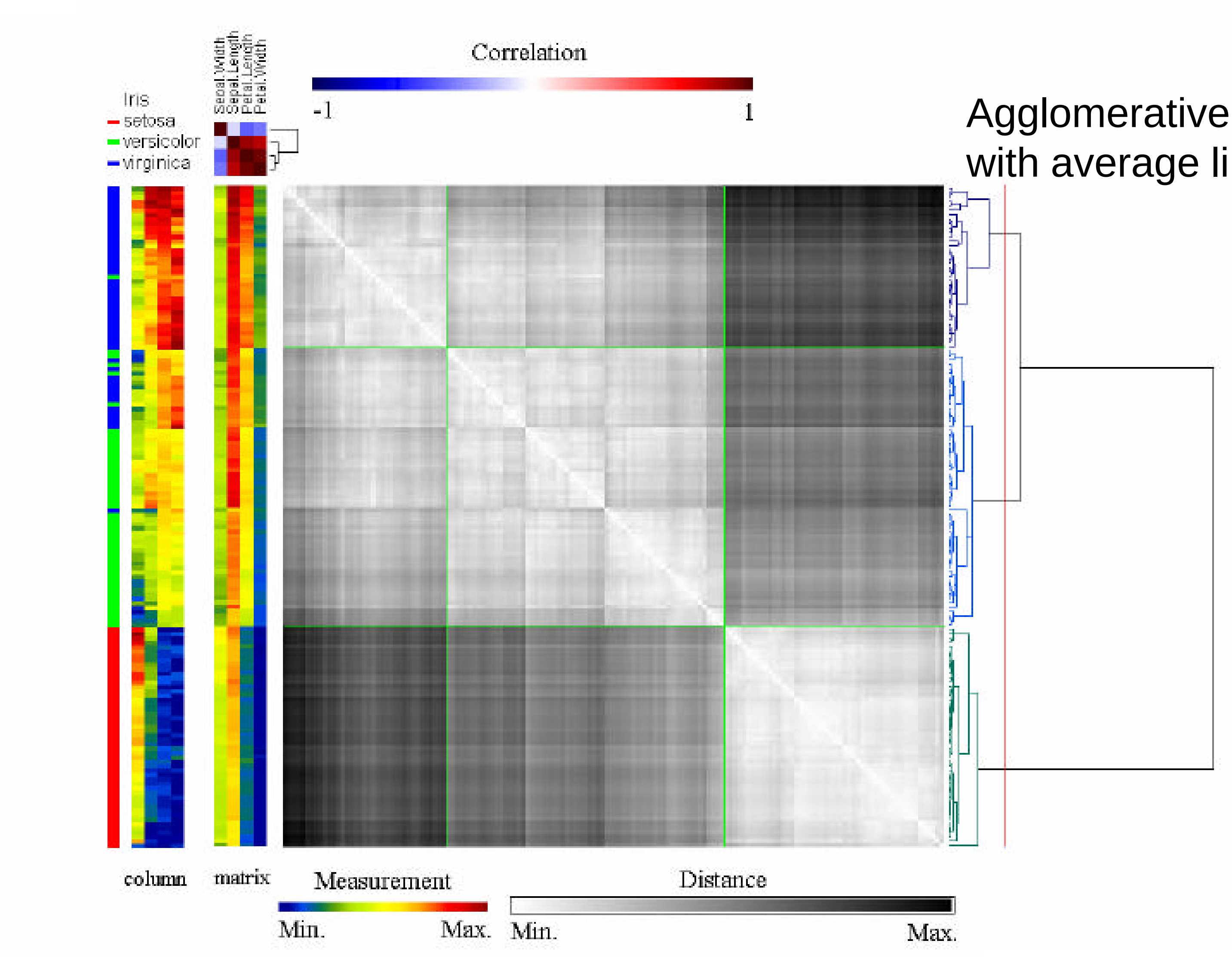
原始資料之呈現
- color spectrum
- variable transformation
- similarity measure




Log2(ratio of gene expression)
(-) down regulated
(0) no differentially expressed
(+) up regulated

Bi-direction colour spectrum for gene expression profile.

GAP: iris example

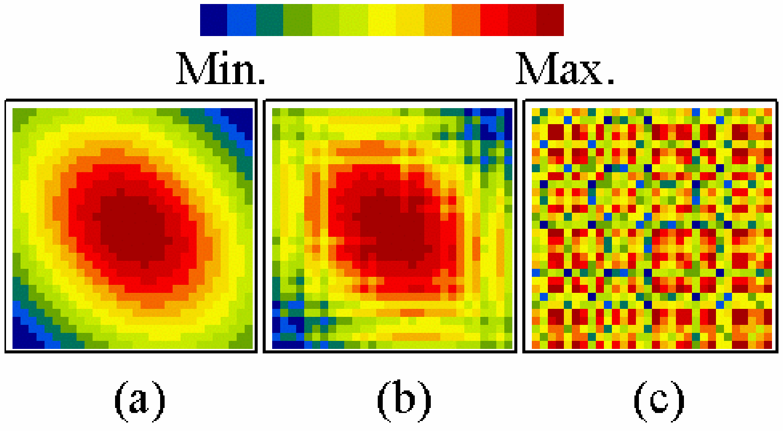


Agglomerative Clustering with average linkage

Seriation

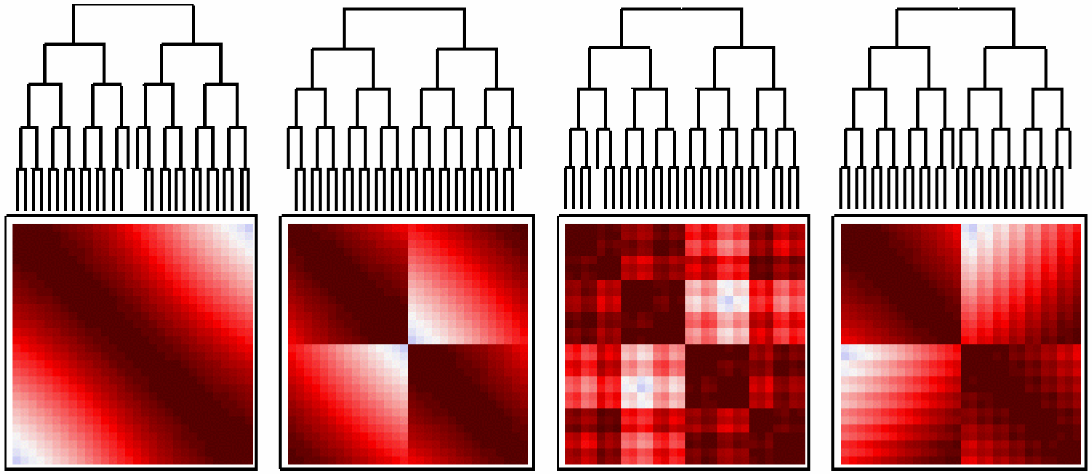
關係矩陣與資料矩陣之排序

- Robinson matrix
- Tree seriation



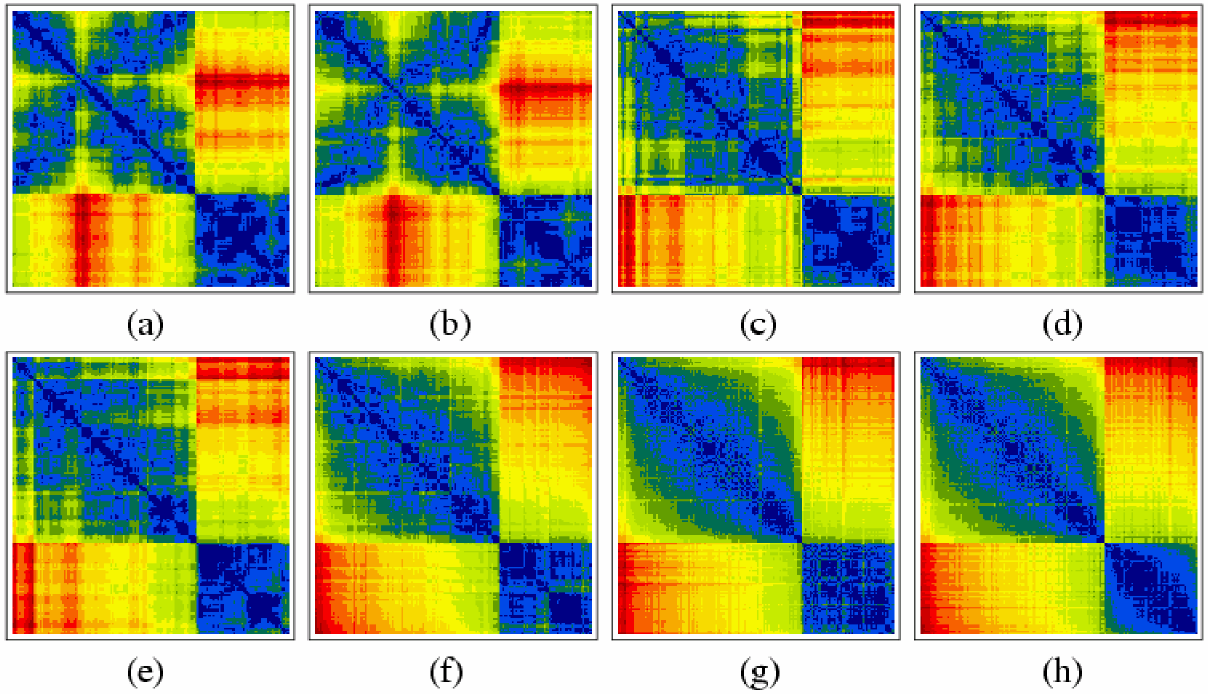
圖三：Robinson 與準-Robinson 矩陣。

Source from Chen (2002).



圖五：不同節點翻轉機制在同一個樹狀結構與關係矩陣上造成的排序差異。

Seriation: iris example



Permuted Euclidean distance maps for Iris data with conventional and seriation algorithms proposed in Chen (2002)
(a) farthest insertion Spanning; (b) nearest insertion spanning;
(c) single linkage cluster tree; (d) complete linkage cluster tree;
(e) average linkage cluster tree; (f) GAP rank-one tree;
(g) GAP rank-two ellipse; (h) GAP rank-1 & 2 double ellipse.

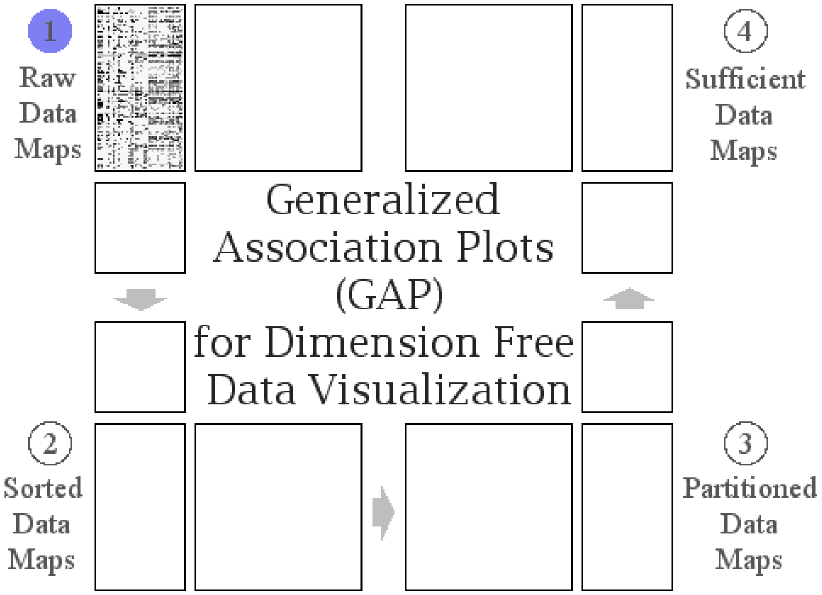
Source from Chen (2002).

More on GAP

Dimension Free
Data Visualization
GAP

Web site: http://gap.stat.sinica.edu.tw/

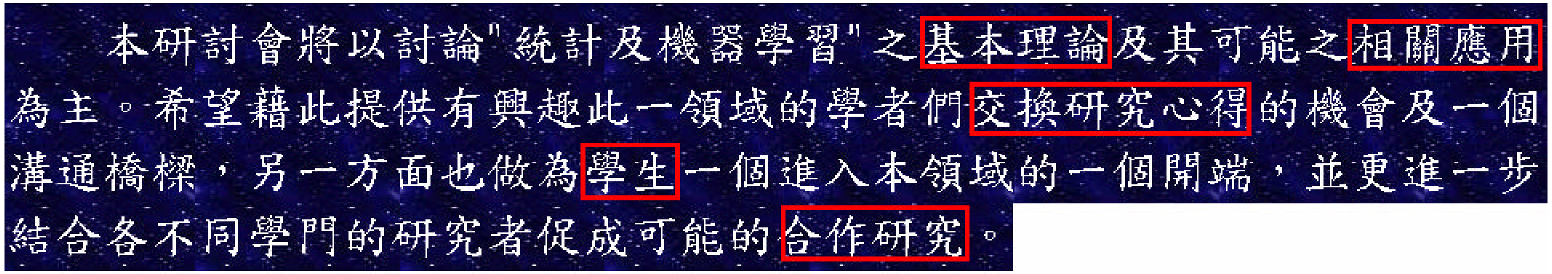
- 類別型 (categorical) 資料之全矩陣視覺化
- 多時點（相同變項）資料之全矩陣視覺化
- 多條件（不同變項）資料之全矩陣視覺化
- 條件式（變項校正）全矩陣視覺化
- 相依 (dependent) 或群集 (clustered) 資料之全矩陣視覺化
- 巨量資料之全矩陣視覺化
- 全矩陣視覺化之遺漏值 (missing value) 處理



Reference

- Dr. Alexander Strehl: http://www.lans.ece.utexas.edu/~strehl/
- Michael Friendly's Home Page:http://www.math.yorku.ca/SCS/friendly.html

- Chen, C. H. (2002), Generalized Association Plots: Information Visualization via Iteratively Generated Correlation Matrices, Statistica Sinica, 12, 7-29.
- Cox, T. F. and Cox, M.A.A. (2001), Multidimensional Scaling, London: Chapman & Hall.
- Hartigan, J. (1972), Direct Clustering of a Data Matrix. Journal of the American Statistical Association, 67(337):123-129.
- Hartigan, J. (1975), Clustering Algorithms, John Wiley and Sons, New York.
- Jacoby, W. G. (1998), Statistical Graphics for Visualizing Multivariate Data, Thousand Oaks, Calif. : Sage Publications.
- Kaufman, L. and Rousseeuw, P.J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York.
- Kohonen, T. (2001), Self-Organizing Maps, Berlin: Springer.
- Minnotte , M. C. and West, R. W., (1999), "The Data Image: a Tool for Exploring High Dimensional Data Sets,". 1998 Proceedings of the ASA Section on Statistical Graphics, in press.



本研討會將以討論"統計及機器學習"之基本理論及其可能之相關應用為主。希望藉此提供有興趣此一領域的學者們交換研究心得的機會及一個溝通橋樑，另一方面也做為學生一個進入本領域的一個開端，並更進一步結合各不同學門的研究者促成可能的合作研究。