Tutorial on Cluster Analysis

Topics

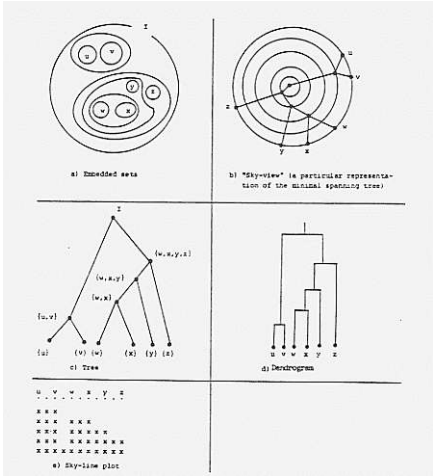
1. An in-depth look at hierarchical clustering, including:
   - Weighting observations
   - Nearest neighbor and reciprocal nearest neighbor algorithms
   - State of the art in complexity
   - Clustering of correspondence analysis factor projections, to bypass normalization problems
2. Graph methods and constrained clustering: these are mostly methods for clustering *on* graphs (as opposed to clustering graphs)
3. Partitioning, distribution mixture modeling with Bayes factors, and Kohonen self-organizing maps, – all of which are based on the EM, expectation-maximization optimization algorithm

Introduction and An Example

Cluster Analysis

Some Terms

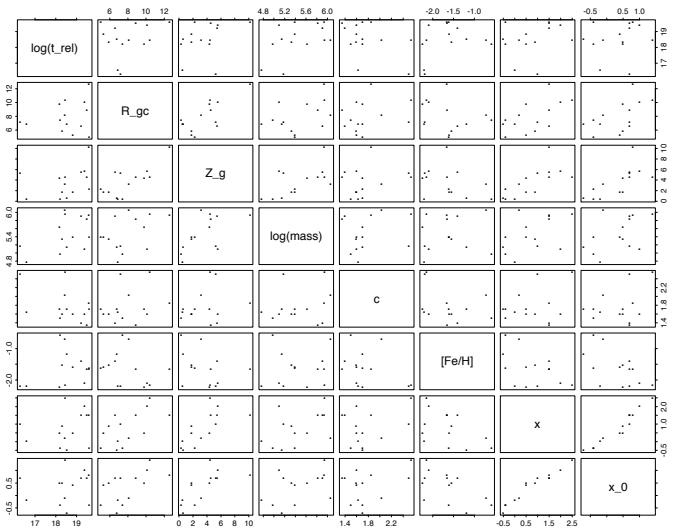
- Unsupervised classification, clustering, cluster analysis, automatic classification. Versus: Supervised classification, discrimant analysis, trainable classifier, machine learning.
- For clustering we can consider (i) partitioning methods, (ii) agglomerative hierarchical classification, (iii) graph methods, (iv) statistical methods, or distribution mixture models, (v) Kohonen self-organizing feature map.
- Then there are combinatorial methods, statistical methods which assume a (data +) noise model, and so on.
- Note that principal components analysis, correspondence analysis, or indeed visualization display methods, can be used for clustering.

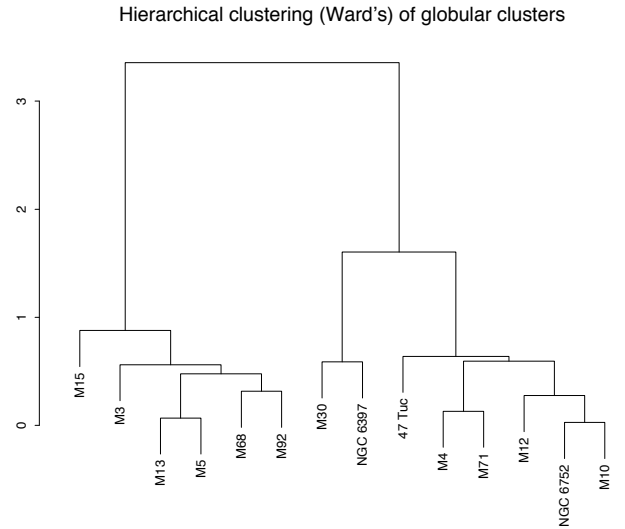


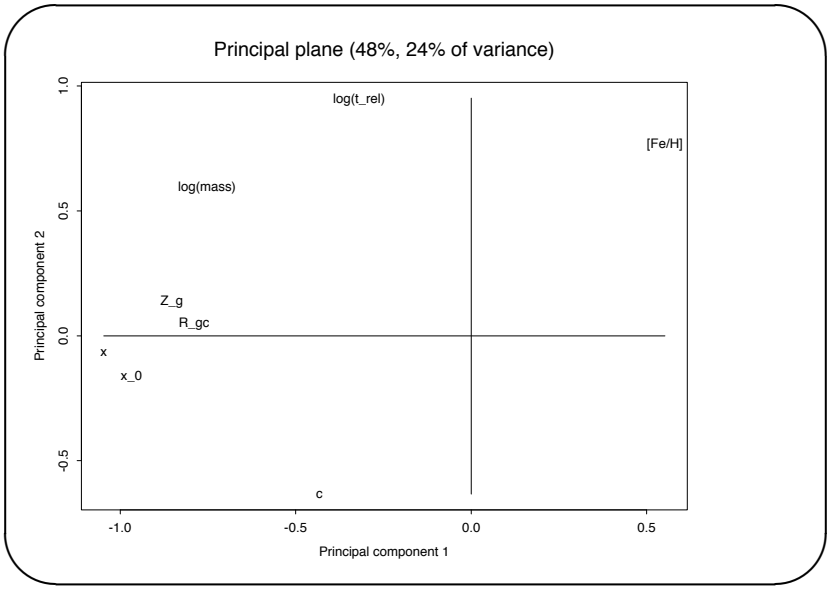
Example: analysis of globular clusters

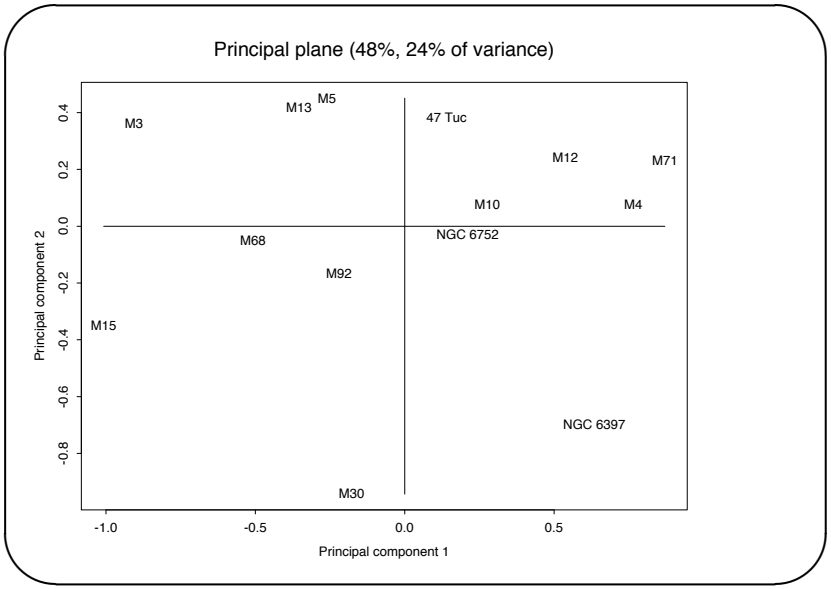
- M. Capaccioli, S. Ortolani and G. Piotto, "Empirical correlation between globular cluster parameters and mass function morphology", AA, 244, 298–302, 1991.
- 14 globular clusters, 8 measurement variables.
- Data collected in earlier CCD (digital detector) photometry studies.
- Pairwise plots of the variables.
- PCA of the variables.
- PCA of the objects (globular clusters).

Object	t_rlx years	Rgc Kpc	Zg Kpc	log(M/ M.)	c	[Fe/H]	x	x0
M15	1.03e+8	10.4	4.5	5.95	2.54	-2.15	2.5	1.4
M68	2.59e+8	10.1	5.6	5.1	1.6	-2.09	2.0	1.0
M13	2.91e+8	8.9	4.6	5.82	1.35	-1.65	1.5	0.7
M3	3.22e+8	12.6	10.2	5.94	1.85	-1.66	1.5	0.8
M5	2.21e+8	6.6	5.5	5.91	1.4	-1.4	1.5	0.7
M4	1.12e+8	6.8	0.6	5.15	1.7	-1.28	-0.5	-0.7
47 Tuc	1.02e+8	8.1	3.2	6.06	2.03	-0.71	0.2	-0.1
M30	1.18e+7	7.2	5.3	5.18	2.5	-2.19	1.0	0.7
NGC 6397	1.59e+7	6.9	0.5	4.77	1.63	-2.2	0.0	-0.2
M92	7.79e+7	9.8	4.4	5.62	1.7	-2.24	0.5	0.5
M12	3.26e+8	5.0	2.3	5.39	1.7	-1.61	-0.4	-0.4
NGC 6752	8.86e+7	5.9	1.8	5.33	1.59	-1.54	0.9	0.5
M10	1.50e+8	5.3	1.8	5.39	1.6	-1.6	0.5	0.4
M71	8.14e+7	7.4	0.3	4.98	1.5	-0.58	-0.4	-0.4









A Formal Definition to Begin With

Hierarchical clustering

- Hierarchical agglomeration on n observation vectors, $i \in I$, involves a series of $1, 2, \ldots, n-1$ *pairwise agglomerations* of observations or clusters, with the following properties.
- A hierarchy $H = \{q | q \in 2^I\}$ such that:
  1. $I \in H$
  2. $i \in H \; \forall i$
  3. for each $q \in H, q' \in H : q \cap q' \neq \emptyset \Longrightarrow q \subset q'$ or $q' \subset q$
- An indexed hierarchy is the pair (H, ν) where the positive function defined on H, i.e., $\nu : H \rightarrow \mathbb{R}^+$, satisfies:
  1. $\nu(i) = 0$ if $i \in H$ is a singleton
  2. $q \subset q' \Longrightarrow \nu(q) < \nu(q')$
- Function ν is the agglomeration level.

- Take $q \subset q'$, let $q \subset q''$ and $q' \subset q''$, and let q'' be the lowest level cluster for which this is true. Then if we define $D(q, q') = \nu(q'')$, D is an ultrametric.
- Recall: **Distances satisfy the triangle inequality** $d(x, z) \leq d(x, y) + d(y, z)$. **An ultrametric satisfies** $d(x, z) \leq \max(d(x, y), d(y, z))$**.** In an ultrametric space triangles formed by any three points are isosceles. An ultrametric is a special distance associated with rooted trees. Ultrametrics are used in other fields also – in quantum mechanics, numerical optimization, number theory, and algorithmic logic.
- In practice, we start with a Euclidean distance or other dissimilarity, use some criterion such as minimizing the change in variance resulting from the agglomerations, and then define $\nu(q)$ as the dissimilarity associated with the agglomeration carried out.

Distance, Similarity, Tree Distance

Metric and Ultrametric

- Triangular inequality:

  Symmetry: $d(a, b) = d(b, a)$

  Positive semi-definiteness: $d(a, b) > 0,$ if $a \neq b; d(a, b) = 0,$ if $a = b$

  Triangular inequality: $d(a, b) \leq d(a, c) + d(c, b)$
- Ultrametric inequality: $d(a, b) \leq \max(d(a, c) + d(c, b))$
- Minkowski metric: $d_p(a, b) = \sqrt[p]{\sum_j \mid a_j - b_j \mid^p} \quad p \geq 1.$
- Particular cases of the Minkowski metric: $p = 2$ gives Euclidean, $p = 1$ gives Hamming or city-block; and $= \infty$ gives $d_\infty(a, b) = max_j \mid a_j - b_j \mid$ which is the "maximum coordinate" or *Chebyshev* distance.
- Also termed L_2, L_1, and L_∞ distances.
- Question: show that squared Euclidean and Hamming distances are the same for binary data.

Metrics

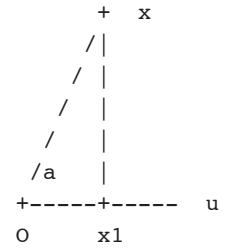
- The notion of distance is crucial, since we want to investigate **relationships** between observations and/or variables.
- Recall: $x = \{3, 4, 1, 2\}, y = \{1, 3, 0, 1\}$, then: scalar product $\langle x, y \rangle = \langle y, x \rangle = x'y = xy' = 3 \times 1 + 4 \times 3 + 1 \times 0 + 2 \times 1.$
- Euclidean norm: $\|x\|^2 = 3 \times 3 + 4 \times 4 + 1 \times 1 + 2 \times 2.$
- Euclidean distance: $d(x, y) = \|x - y\|$. The squared Euclidean distance is: $3 - 1 + 4 - 3 + 1 - 0 + 2 - 1$
- Orthogonality: x is orthogonal to y if $\langle x, y \rangle = 0.$
- Distance is symmetric, $d(x, y) = d(y, x)$; positive, $d(x, y) \geq 0$; and definite, $d(x, y) = 0 \Longrightarrow x = y.$

Metrics (cont'd.)

- Any symmetric, positive, definite matrix M defines a generalized Euclidean space. Scalar product is $\langle x, y \rangle_M = x'My$, norm is $\|x\|^2 = x'Mx$, and Euclidean distance is $d(x, y) = \|x - y\|_M$.
- Classical case: $M = I_n$, the identity matrix.
- Normalization to unit variance: M is diagonal matrix with ith diagonal term $1/\sigma_i^2$.
- Mahalanobis distance: M is inverse variance-covariance matrix.
- Next topic: Scalar product defines orthogonal projection.

Metrics (cont'd.)

- Projected value, projection, coordinate: $x_1 = (x'Mu/u'Mu)u$. Here x_1 and u are both vectors.
- Norm of vector $x_1 = (x'Mu/u'Mu)\|u\| = (x'Mu)/\|u\|$.
- The quantity $(x'Mu)/(\|x\|\|u\|)$ can be interpreted as the cosine of the angle a between vectors x and u.

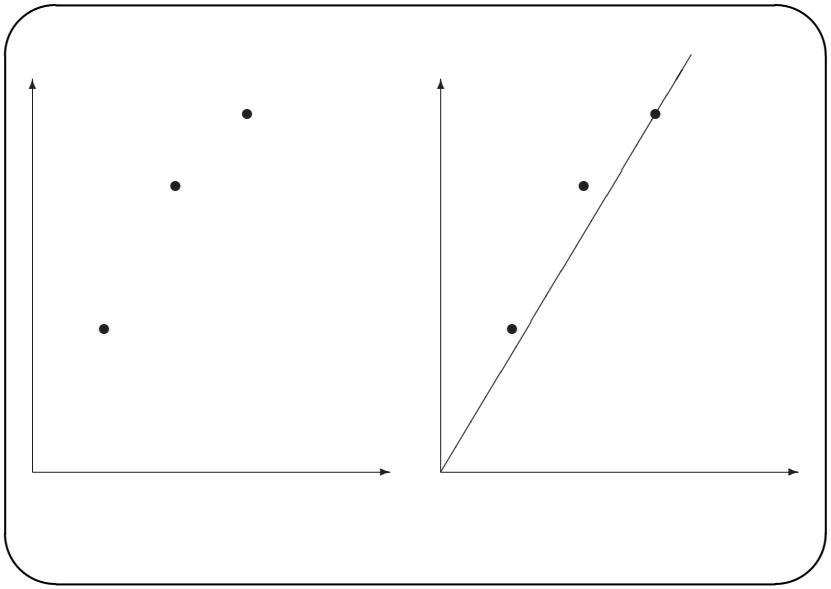


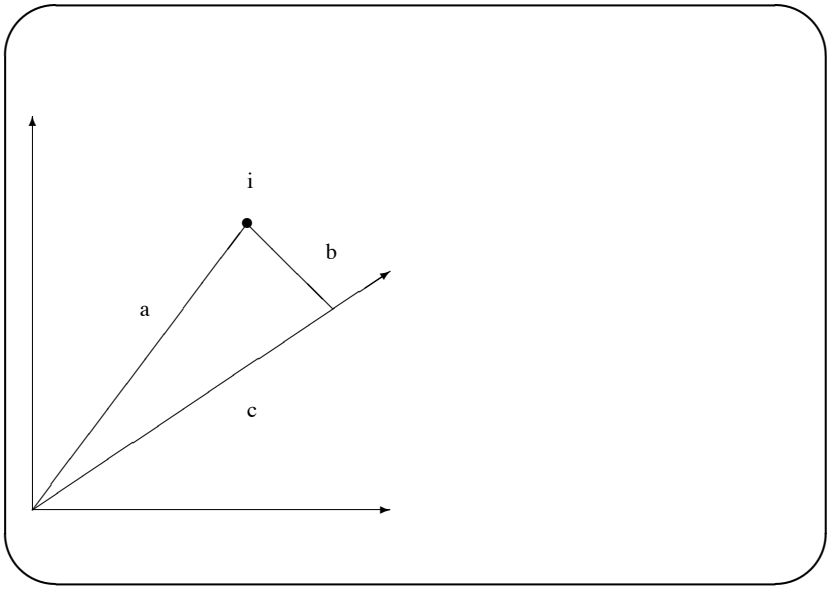
Least Squares Optimal Projection of Points

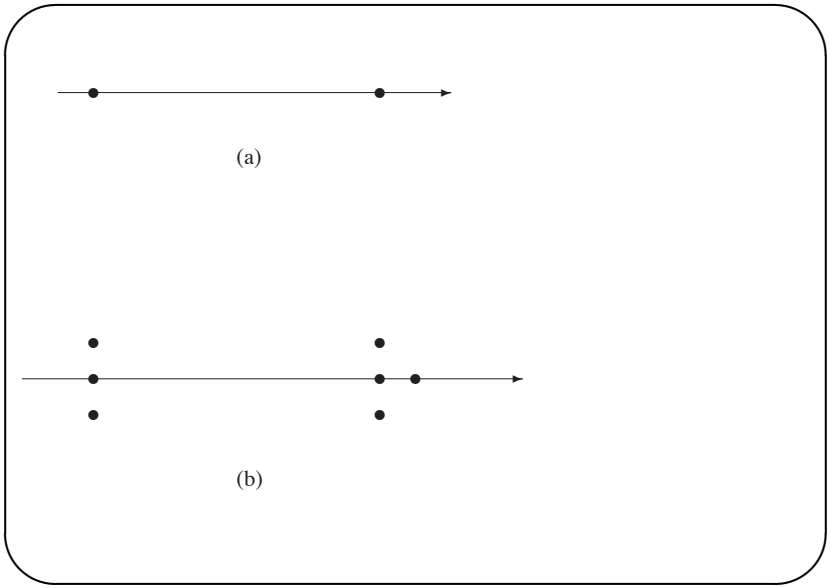
- Plot of 3 points in $\mathbb{R}^2$ (see following slides).
- PCA: determine best fitting axes.
- Examples follow.
- Note: optimization means either (i) closest axis to points, or (ii) maximum elongation of projections of points on the axis.
- This follows from Pythagoras's theorem: $x^2 + y^2 = z^2$. Call z the distance from the origin to a point. Let x be the distance of the projection of the point from the origin. Then y is the perpendicular distance from the axis to to the point.
- Minimizing y is the same as maximizing x (because z is fixed).

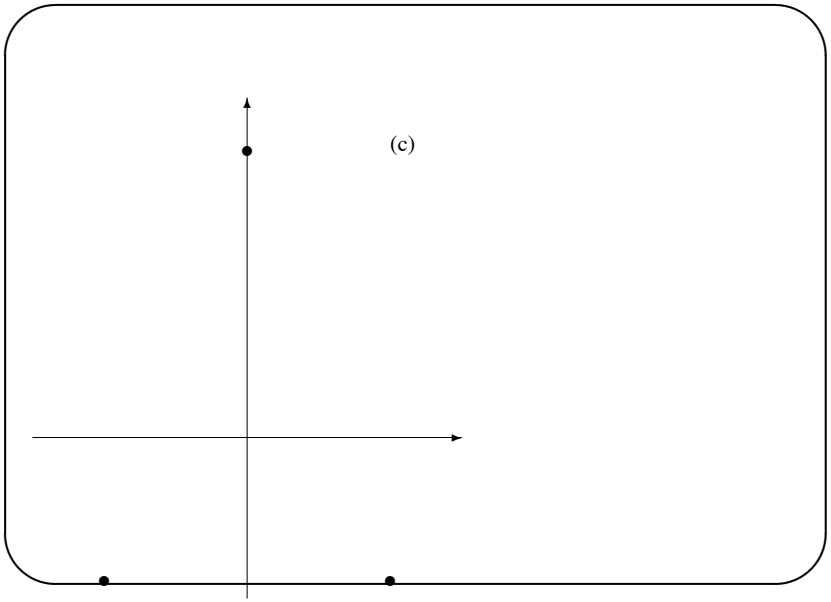
Examples of Optimal Projection

$$\begin{pmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 5 \end{pmatrix}$$




i
b
a
c


(a)
(b)


(c)

Cosine Coefficient (cf. Principal Components Analysis)

- The projection of vector $\mathbf{x}$ onto axis $\mathbf{u}$ is $\mathbf{y} = \frac{\mathbf{x}'M\mathbf{u}}{\|\mathbf{u}\|_M}\ \mathbf{u}$
- I.e. the coordinate of the projection on the axis is $\mathbf{x}'M\mathbf{u}/\|\mathbf{u}\|_M$.
- This becomes $\mathbf{x}'M\mathbf{u}$ when the vector $\mathbf{u}$ is of unit length.
- The cosine of the angle between vectors $\mathbf{x}$ and $\mathbf{y}$ in the usual Euclidean space is $\mathbf{x}'\mathbf{y}/\|\mathbf{x}\|\|\mathbf{y}\|$.
- That is to say, we make use of the triangle whose vertices are the origin, the projection of $\mathbf{x}$ onto $\mathbf{y}$, and vector $\mathbf{x}$.
- The cosine of the angle between $\mathbf{x}$ and $\mathbf{y}$ is then the coordinate of the projection of $\mathbf{x}$ onto $\mathbf{y}$, divided by the – hypotenuse – length of $\mathbf{x}$.
- The correlation coefficient between two vectors is then simply the cosine of the angle between them, when the vectors have first been centred (i.e. $\mathbf{x} - \mathbf{g}$ and $\mathbf{y} - \mathbf{g}$ are used, where $\mathbf{g}$ is the overall centre of gravity.

Normalization $\Longrightarrow$ Scalar Product gives Correlation

- Let r_{ij} be the original measurements.
- Then define: $x_{ij} = \frac{r_{ij} - \overline{r}_j}{s_j\sqrt{n}}$
- $\overline{r}_j = \frac{1}{n}\sum_{i=1}^n r_{ij}$
- $s_j^2 = \frac{1}{n}\sum_{i=1}^n (r_{ij} - \overline{r}_j)^2$
- Then the matrix to be diagonalized in PCA, or the all-pairwise scalar products of observation vectors, is of $(j,k)^{th}$ term:
  $\rho_{jk} = \sum_{i=1}^n x_{ij}x_{ik} = \frac{1}{n}\sum_{i=1}^n (r_{ij} - \overline{r}_j)(r_{ik} - \overline{r}_k)/s_j s_k$
- This is the correlation coefficient between variables j and k.
- Have distance
  $d^2(j,k) = \sum_{i=1}^n (x_{ij} - x_{ik})^2 = \sum_{i=1}^n x_{ij}^2 + \sum_{i=1}^n x_{ik}^2 - 2\sum_{i=1}^n x_{ij}x_{ik}$
- First two terms both yield 1. Hence:

- $d^2(j,k) = 2(1 - \rho_{jk})$
- Thus the distance between variables is directly proportional to the correlation between them.
- For row points (objects, observations):
  $d^2(i,h) = \sum_j (x_{ij} - x_{hj})^2 = \sum_j (\frac{r_{ij} - r_{hj}}{\sqrt{n}s_j})^2 = (\mathbf{r}_i - \mathbf{r}_h)'M(\mathbf{r}_i - \mathbf{r}_h)$
- $\mathbf{r}_i$ and $\mathbf{r}_h$ are column vectors (of dimensions $m \times 1$) and M is the $m \times m$ diagonal matrix of j^{th} element $1/ns_j^2$.
- Therefore d is a Euclidean distance associated with matrix M.
- Note that the row points are now centred but the column points are not: therefore the latter may well appear in one quadrant on output listings.

Cosine, Correlation Coeffs. Now: Further Examples of Similarities

- Jaccard coefficient for binary vectors $\mathbf{a}$ and $\mathbf{b}$. N is counting operator:
  $s(a,b) = \frac{N_j(a_j=b_j=1)}{N_j(a_j=1)+N_j(b_j=1)-N_j(a_j=b_j=1)}$
- Jaccard similarity coefficient of vectors (10001001111) and (10101010111) is $5/(6+7-5) = 5/8$. In vector notation: $s(a,b) = \frac{\mathbf{a'b}}{\mathbf{a'a}+\mathbf{b'b}-\mathbf{a'b}}$.
- Jaccard coefficient uses counts of presence/absences in cross-tabulation of binary presence/absence vectors:

	a/present	a/absent
b/present	n1	n2
b/absent	n3	n4

- A number of such measures have been used in information retrieival, or numerical taxonomy: Jaccard, Dice, Tanimoto, ...

Upstream of Distances or Similarities: Data Coding

Record x:	S1, 18.2, X
Record y:	S1, 6.7, —

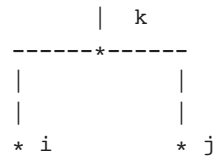
Two records (x and y) with three variables (Seyfert type, magnitude, X-ray emission) showing disjunctive coding.

	Seyfert type spectrum				Integrated magnitude		X–ray data?
	S1	S2	S3	—	≤ 10	> 10	Yes
x	1	0	0	0	0	1	1
y	1	0	0	0	1	0	0

Concluding for the present on Distances

- A distance, as seen, is defined on a set of objects $\mathbf{x}$, as a mapping $d : \mathbf{x} \times \mathbf{x} \longrightarrow \mathbb{R}^+$, where the result (right hand term) is a value in the set of positive reals.
- Alternatively expressed, for $x_i, x_j \in \mathbf{x}$, then $d(x_i, x_j) \in \mathbb{R}^+$.
- A Euclidean space is a particular metric space. If we allow for infinite dimension, then this is termed a Hilbert space.
- Euclidean distance is defined from scalar product. Scalar product gives cosine of angle between two vectors. If vectors are suitably normalized, then we have correlations between them. A more "global" normalization is involved when we modify the Euclidean distance to give the Mahalanobis distance.

Some Properties of Ultrametrics

- Distance defined strictly *on a tree*.
- 

  Considering 3 points, i, j, k we have already considered the relationship $d_{xy} \leq \max\{d_{xz}, d_{yz}\}$ where x, y, z take on the different values i, j, k in any order.
- Furthermore: any triangle, formed from a triplet of points, must be equilateral, or isosceles with small base.
- Topologically, every open ball is also a closed ball. We term this a *clopen* ball.
- Every point in a (closed or open) ball can be taken as its center.

- The radius of a ball is identical to its diameter.
- If two (either both open or both closed) balls are overlapping, then one must be enclosed in the other.
- *Conclude: an ultrametric, or tree or hierarchic distance, is very peculiar!*

A Worked Example of Hierarchical Agglomerative clustering

Note: the agglomerative criterion used is very important.

Single Linkage Hierarchical Clustering

Dissimilarity matrix defined for 5 objects

	1	2	3	4	5
1	0	4	9	5	8
2	4	0	6	3	6
3	9	6	0	6	3
4	5	3	6	0	5
5	8	6	3	5	0

Agglomerate 2 and 4 at dissimilarity 3

	1	2U4	3	5
1	0	4	9	8
2U4	4	0	6	5
3	9	6	0	3
5	8	5	3	0

Agglomerate 3 and 5 at dissimilarity 3

Single Linkage Hierarchical Clustering – 2

	1	2U4	3U5
1	0	4	8
2U4	4	0	5
3U5	8	5	0

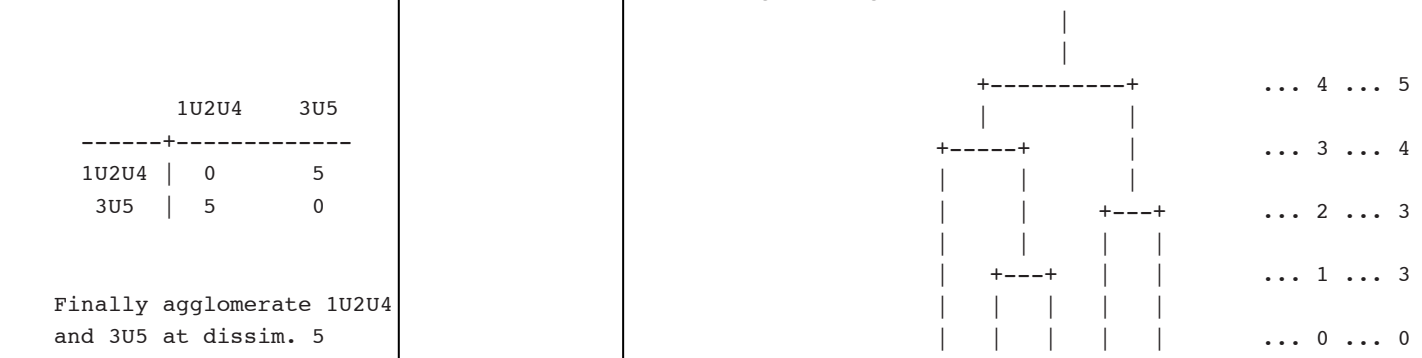
Agglomerate 1 and 2U4 at dissimilarity 4

	1U2U4	3U5
1U2U4	0	5
3U5	5	0

Finally agglomerate 1U2U4 and 3U5 at dissim. 5

Single Linkage Hierarchical Clustering – 3

Resulting dendrogram



r = ranks or levels. c = criterion values (linkage wts).

Single Linkage Hierarchical Clustering – 3

Input An $n(n-1)/2$ set of dissimilarities.

Step 1 Determine the smallest dissimilarity, d_{ik}.

Step 2 Agglomerate objects i and k: i.e. replace them with a new object, $i \cup k$; update dissimilarities such that, for all objects $j \neq i, k$:

$$d_{i \cup k, j} = \min \{d_{ij}, d_{kj}\}.$$

Delete dissimilarities d_{ij} and d_{kj}, for all j, as these are no longer used.

Step 3 While at least two objects remain, return to Step 1.

Single Linkage Hierarchical Clustering – 4

- Precisely $n-1$ levels for n objects. Ties settled arbitrarily.
- Note single linkage criterion.
- Disadvantage: chaining. "Friends of friends" in the same cluster.
- Lance-Williams cluster update formula:
  $d(i \cup j, k) = \alpha_i d(i,k) + \alpha_j d(j,k) + \beta d(i,j) + \gamma \mid d(i,k) - d(j,k) \mid$ where coefficients α_i, α_j, β, and γ define the agglomerative criterion.
- For single link, $\alpha_i = 0.5$, $\beta = 0$ and $\gamma = -0.5$.
- These values always imply: $\min\{d_{ik}, d_{jk}\}$
- Ultrametric distance, δ, resulting from the single link method is such that $\delta(i,j) \leq d(i,j)$ always. It is also unique (with the exception of ties). So single link is also termed the *subdominant ultrametric* method.

Remarks on Hierarchical Clustering Criteria

- Complete link: substitute max for min in single link.
- Complete link leads to compact clusters.
- Single link defines the cluster criterion from the closest object in the cluster. Complete link defines the cluster criterion from the furthest object in the cluster.
- Single link yields the *maximal inferior ultrametric*, or *subdominant ultrametric*.
- What this means is: let δ_{ij} be an ultrametric distance derived from the single link hierarchy, and let d_{ij} be the original corresponding distance. Then $\delta_{ij} \leq d_{ij}$, and δ_{ij} is the best such fit to d_{ij} "from below". This *subdominant utrametric* is unique.
- Analogously, complete link yields a *minimal superior ultrametric*. However this is not unique.
- Robin Sibson developed an $O(n^2)$ algorithm for single link.

R. Sibson, "SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method", Computer Journal, 16, 30-34, 1973.
Note here: *optimal*, i.e. $O(n^2)$.

Robin Sibson was Professor of Statistics at the University of Bath, and later Vice-Chancellor of the University of Kent at Canterbury. In 2000, he became Chief Executive of the Higher Education Statistics Agency, HESA, in the UK.

- Daniel Defays developed an $O(n^2)$ algorithm for a complete link method. D. Defays, "An efficient algorithm for a complete link method", Computer Journal, 20, 364-366, 1977.

  Daniel Defays went on to work also in official statistics, in Eurostat, the Statistical Office of the European Union.
- Other criteria define $d(i \cup j, k)$ from the distance between k and something closer to the *mean or center* of i and j. These criteria include the median, centroid and minimum variance methods.

Remarks on Hierarchical Clustering Criteria (Cont'd.)

- A problem that can arise: inversions in the hierarchy. I.e. the cluster criterion value is not monotonically increasing. That leads to cross-overs in the dendrogram.
- Of the above agglomerative methods, the single link, complete link, and minimum variance methods can be shown to never allow inversions. They satisfy the *reducibility property*.

  First formulated by Michel Bruynooghe, working in Benzécri's lab in the late 1970s. Bruynooghe now works in a university group on photonic systems in Strasbourg, France.
- We will return to this property – which guarantees no inversions or monotonic behavior in the sequence of agglomerations – later when we discuss representation or display aspects of hierarchies.

Summary of Hierarchical Agglomerative Criteria

Note: we should distinguish clearly between clustering method (implying a stepwise optimization criterion) and an algorithm.

N. Jardine and R. Sibson, Mathematical Taxonomy, Wiley, 1971, p. 42

Hierarchical clustering methods (and aliases).	Lance and Williams dissimilarity update formula.	Coordinates of centre of cluster, which agglomerates clusters i and j.	Dissimilarity between cluster centres g_i and g_j.
Single link (nearest neighbor).	$\alpha_i = 0.5$ $\beta = 0$ $\gamma = -0.5$ (More simply: $min\{d_{ik}, d_{jk}\}$)		
Complete link (diameter).	$\alpha_i = 0.5$ $\beta = 0$ $\gamma = 0.5$ (More simply: $max\{d_{ik}, d_{jk}\}$)		
Group average (average link, UPGMA).	$\alpha_i = \frac{\lvert i\rvert}{\lvert i\rvert+\lvert j\rvert}$ $\beta = 0$ $\gamma = 0$		

Hierarchical clustering methods (and aliases).	Lance and Williams dissimilarity update formula.	Coordinates of centre of cluster, which agglomerates clusters i and j.	Dissimilarity between cluster centres g_i and g_j.
Median method (Gower's, WPGMC).	$\alpha_i = 0.5$ $\beta = -0.25$ $\gamma = 0$	$\mathbf{g} = \frac{\mathbf{g}_i+\mathbf{g}_j}{2}$	$\lVert\mathbf{g}_i - \mathbf{g}_j\rVert^2$
Centroid (UPGMC).	$\alpha_i = \frac{\lvert i\rvert}{\lvert i\rvert+\lvert j\rvert}$ $\beta = -\frac{\lvert i\rvert\lvert j\rvert}{(\lvert i\rvert+\lvert j\rvert)^2}$ $\gamma = 0$	$\mathbf{g} = \frac{\lvert i\rvert\mathbf{g}_i+\lvert j\rvert\mathbf{g}_j}{\lvert i\rvert+\lvert j\rvert}$	$\lVert\mathbf{g}_i - \mathbf{g}_j\rVert^2$
Ward's method (minimum variance, error sum of squares.	$\alpha_i = \frac{\lvert i\rvert+\lvert k\rvert}{\lvert i\rvert+\lvert j\rvert+\lvert k\rvert}$ $\beta = -\frac{\lvert k\rvert}{\lvert i\rvert+\lvert j\rvert+\lvert k\rvert}$ $\gamma = 0$	$\mathbf{g} = \frac{\lvert i\rvert\mathbf{g}_i+\lvert j\rvert\mathbf{g}_j}{\lvert i\rvert+\lvert j\rvert}$	$\frac{\lvert i\rvert\lvert j\rvert}{\lvert i\rvert+\lvert j\rvert}\lVert\mathbf{g}_i - \mathbf{g}_j\rVert^2$

Observation Weighting

- Note how centroid and Ward's minimum variance methods allow for a simple but satisfactory way to weight the observatations:
- New cluster center: $q'' = (m_q q + m_{q'} q')/(m_q + m_{q'})$.
- Dissimilarity between new cluster center is $(m_q m_{q'})/(m_q + m_{q'})\|q - q'\|^2$.
- Typically, $m_q = m_{q'} = 1/n$ to begin with, where we have n observations.
- To weight observations, just take these weights as other than identical and constant.
- Our software – in C, Java and R – supports observation weighting. (Of course there is no problem with identical, constant weights.)

Basic or Traditional Algorithms

Agglomerative Algorithm Based on Data

Step 1 Examine all interpoint dissimilarities, and form cluster from two closest points.

Step 2 Replace two points clustered by representative point (centre of gravity) or by cluster fragment.

Step 3 Return to Step 1, treating clusters as well as remaining objects, until all objects are in one cluster.

Agglomerative Algorithm Based on Dissimilarities

Step 1 Form cluster from smallest dissimilarity.

Step 2 Define cluster; remove dissimilarity of agglomerated pair. Update dissimilarities from cluster to all other clusters/singletons.

Step 3 Return to Step 1, treating clusters as well as remaining objects, until all objects are in one cluster.

Computational Complexity

- Find closest dissimilarity in order to carry out an agglomeration: take each observation and match (Euclidean distance etc.) with every other. We take n observations, and we carry out $O(n)$ matchings. So complexity is $O(n^2)$. We repeat this for $n-1$ agglomerations. So complexity overall is $O(n^3)$.
- Say we have dissimilarities. (These could well be distances; or *mutatis mutandis* similarities.) All pairwise dissimilarities are needed. (Not precluding an upper, or lower, half matrix of dissimilarities.) So, to set up, the complexity is $O(n^2)$. Now we find the minimum dissimilarity, taking $O(n^2)$ effort to scan all dissimilarities. We agglomerate and update our dissimilarity matrix (again, $O(n^2)$ effort). So far, everything together is of $O(n^2)$ effort. We repeat this procedure $n-1$ times. All told, complexity is $O(n^3)$.

Minimum Variance Method

Minimum variance agglomeration

- For Euclidean distance inputs, the following definitions hold for the minimum variance or Ward error sum of squares agglomerative criterion.
- Coordinates of the new cluster center, following agglomeration of q and q', where m_q is the mass of cluster q defined as cluster cardinality, and (vector) q denotes sing overloaded notation the center of (set) cluster q: $q'' = (m_q q + m_{q'} q')/(m_q + m_{q'})$.
- Following the agglomeration of q and q', we define the following dissimilarity: $(m_q m_{q'})/(m_q + m_{q'})\|q - q'\|^2$.
- Hierarchical clustering is usually based on factor projections, if desired using a limited number of factors (e.g. 7) in order to filter out the most useful information in our data. (See discussion later.)
- In such a case, hierarchical clustering can be seen to be a mapping of Euclidean distances into ultrametric distances.

Minimum variance method: properties

- We seek to agglomerate two clusters, c_1 and c_2, into cluster c such that the within-class variance of the partition thereby obtained is minimum.
- Alternatively, the between-class variance of the partition obtained is to be maximized.
- Let P and Q be the partitions prior to, and subsequent to, the agglomeration; let $p_1, p_2, \ldots$ be classes of the partitions.

$$P = \{p_1, p_2, \ldots, p_k, c_1, c_2\}$$
$$Q = \{p_1, p_2, \ldots, p_k, c\}.$$

- Total variance of the cloud of objects in m-dimensional space is decomposed into the sum of within-class variance and between-class variance. This is Huyghen's theorem in classical mechanics.
- Total variance, between-class variance, and within-class variance are as follows:

$V(I) = \frac{1}{n}\sum_{i\in I}(i-g)^2$, $V(P) = \sum_{p\in P}\frac{|p|}{n}(p-g)^2$; and $\frac{1}{n}\sum_{p\in P}\sum_{i\in p}(i-p)^2$.

- For two partitions, before and after an agglomeration, we have respectively:

$$V(I) = V(P) + \sum_{p\in P} V(p)$$

$$V(I) = V(Q) + \sum_{p\in Q} V(p)$$

- From this, it can be shown that the criterion to be optimized in agglomerating c_1 and c_2 into new class c is:

$$\begin{aligned} V(P) - V(Q) &= V(c) - V(c_1) - V(c_2) \\ &= \frac{|c_1|\ |c_2|}{|c_1|+|c_2|}\|\mathbf{c_1} - \mathbf{c_2}\|^2 \ , \end{aligned}$$

Reciprocal Nearest Neigbors, NN-Chains

Efficient NN chain algorithm

- A *NN*-chain (nearest neighbor chain)

Efficient NN chain algorithm (cont'd.)

- An *NN*-chain consists of an arbitrary point followed by its *NN*; followed by the *NN* from among the remaining points of this second point; and so on until we necessarily have some pair of points which can be termed reciprocal or mutual *NN*s. (Such a pair of *RNN*s may be the first two points in the chain; and we have assumed that no two dissimilarities are equal.)
- In constructing a *NN*-chain, irrespective of the starting point, we may agglomerate a pair of *RNN*s as soon as they are found.
- Exactness of the resulting hierarchy is guaranteed when the cluster agglomeration criterion respects the *reducibility property*.
- Inversion impossible if: $d(i,j) < d(i,k)$ or $d(j,k) \Rightarrow d(i,j) < d(i \cup j, k)$

NN-Chain Algorithm Complexity – for "Geometric" Methods

- Firstly, take observation points in space, starting with an arbitrary point. Find its NN; and latter's NN; and latter's; ... until we have RNN. Each such operation is called a *growth*. Agglomerate. Such an operation is called a *contraction*. Restart process from *last point of NN-chain, before the RNN pair*. The number of *contractions* is necessarily $n - 1$. The number of *growths* cannot exceed $3n - 3$. (Why? Because we have n points to begin with; we have $n - 1$ cluster points created; and we have $n - 1$ "stub" points to consider which allow an RNN pair to be created from the final link in the NN-chain. Total upper bounded by: $3n - 3$.)
- So the total number of growths and contractions is linear in n, i.e. is $O(n)$. Now each growth is based on a NN search, hence $O(n)$. Overall, the complexity is $O(n^2)$.
- Storage here is the original data and cluster points, hence $O(n)$.

NN-Chain Algorithm Complexity – for "Graph" Methods

- Start from dissimilarity matrix, $O(n^2)$ to create. Storage here is bounded by the dissimilarity data, hence $O(n^2)$.
- After each agglomeration, keep the dissimilarity matrix updated. $O(n)$ effort required at each agglomeration, since we use Lance-Williams on 2 rows and on 2 columns of the dissimilarity matrix. Note that the dissimilarity matrix has numbers of rows and columns that are less 1 at each step.
- There are, in all, $n - 1$ agglomerations. So all updates to the dissimilarity matrix are $O(n)$. Each such update taking $O(n)$ implies overall $O(n^2)$ effort.
- What about the growths? Just like before, the total number of NN-chain growths is $O(n)$. Each such growth requires $O(n)$ effort because we just have to scan one row (or one column since dissimilarities are assumed symmetric).
- We see that overall complexity is $O(n^2)$.

NN-Chain Algorithm Complexity – for "Graph" Methods

- There is enormous confusion in the literature about this result!
- Confusion is most often about complete link method.
- Edward Fox, Virginia Tech, ei.cs.vt.edu/∼cs5604/f95/cs5604cnCL/CL-alg-details.html
  "Complete link: Time: Voorhees alg. worst case is O(N**3)
  Implementations of the general algorithm:
  - Stored matrix approach: Use matrix, and then apply Lance-Williams to recalculate dissimilarities between cluster centers. Storage is therefore O(N**2) and time is at least O(N**2), but will be O(N**3) if matrix is scanned linearly.
  - Stored data approach: O(N) space for data but recompute pairwise dissimilarities so need O(N**3) time
  - Sorted matrix approach: O(N**2) to calculate dissimilarity matrix, O(N**2 log N**2) to sort it, O(N**2) to construct hierarchy, but one need not store the data set, and the matrix can be processed linearly, which reduces disk accesses."
- Hinrich Schütze, Stuttgart, www-csli.stanford.edu/∼schuetze/completelink.html
  "The worst case time complexity of complete-link clustering is at most $O(n^2 \log n)$.
  (My intuition is that complete link clustering is easier than sorting a set of n^2 numbers, so there should be a more efficient algorithm. Let me know if you know of one!)"

- Peter Scheuermann, Northwestern, www.ece.northwestern.edu/∼peters/publications/euro_par.pdf
  M. Dash, S. Petrutiu and P. Scheuermann, Efficient Parallel Hierarchical Clustering, Proc. 10th International Euro-Par Conference, Italy, September 2004, LNCS 3149, pp. 363-371.
  "Existing algorithms take $O(N^2 \log N)$ CPU time and require (N^2) memory."
- David Eppstein, UCI, www.ics.uci.edu/∼eppstein/280/tree.html
  "However, Neighbor-Joining seems more difficult, with the best known time bound being $O(n^3)$ (and some commonly available implementations taking even more than that)."
- Confusion reigns! But $O(n^2)$ time algorithms ("optimal" as termed by Sibson) have been known and implemented (e.g. in David Wishart's CLUSTAN package, since 1984), since the early 1980s. There is no excuse for not knowing this!

Parallel RNN Agglomerations

The problem for parallel algorithms is that there can be many "stub" points b.

Parallel RNN Algorithm Complexity

- (Re)determine all NNs and RNNs.
- Agglomerate all RNNs, replacing each with cluster point.
- Repeat.
- This works well if our data is uniformly distributed. But, eh..., if we assume clustering in our data, then our starting point is that the data are *not* uniformly distributed.
- Analysis: Say we find r_1 RNNs the first time around, so $n - r_1$ points remain. Have: $1 \leq r_1 \leq \lfloor n/2 \rfloor$.
- To begin with we have n NN calculations.
- Next step, we have $n - r_1$ NN calculations.
- Next: $n - r_1 - r_2$ NN calculations.
- Etc. such that $r_1 + r_2 + r_3 + \ldots + 1 = n - 1$
- If we assume $O(n^2)$ for each step, then overall $O(n^3)$.
- F. Murtagh, "Complexities of hierarchic clustering algorithms: state of the art", Comp. Stat. Quart., 1, 101–113, 1984:

- For median and centroid methods, where clusterwise dissimilarities remain Euclidean, then the number of points which can simultaneously have a given point as NN in m-dimensional Chebyshev space is $3^m - 1$, i.e. the number of cubes which are adjacent to a given cube. Constant for given m. We assume that this bound holds for Euclidean case also.
- For these methods, then, we have time $O(n^2)$.
- Result also established by W.H.E. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods", Jnl. Classification, 1, 7–24, 1984. Based on sphere packing (and Hadwiger numbers).
- Note: this $O(n^2)$ computational worst case holds for the centroid and median methods only, viz. where clusterwise dissimilarities remain Euclidean.

RNNs and NN-Chain Algorithm Complexity – Notes

- We have ignored dimensionality, m. Complexity of foregoing algorithms is linear in m, $O(m)$.
- We have seen in all algorithms that the complexity is based on a NN-search requirement. For an NN search, the latter is $O(n)$ in general.
- But if we can speed up NN-search, then we have a way to break the overall $O(n^2)$ computational complexity barrier.
- The first "data storage" RNN-chain based algorithm works for any "geometric" clustering algorithm. The second "dissimilarity storage" RNN-chain based algorithm works for any "graph" clustering algorithm.
- The RNN and NN-chain algorithms require the reducibility property so that inversions don't materialize.
- So okay are: Ward's minimum variance; weighted and unweighted average

linkage methods – UPGMA, WPGMA.

- But *not* okay are: centroid, median.
- If we can expedite NN calculations then we can be even better. In fact, we can achieve $O(n)$ performance whenever each NN calculation is constant, or $O(1)$.
- There are ways to allow for fast NN finding using approaches taken from computational geometry.
- And work by JL Bentley and JH Friedman around 1977–1978 showed that, for uniformly distributed data in a bounded region, the NN of a point could be found in constant expected time.
- Rohlf in 1977 and 1978 developed an $O(n \log \log n)$ expected time algorithm for the single link method and extended it to the centroid and median methods.
- Various experimental results to about $n = 12000$ in F Murtagh, "Expected-time complexity results for hierarchic clustering algorithms which use cluster centres", Information Processing Letters, 16, 237–241, 1983.

RNNs and NN-Chain Algorithm Complexity – Summary

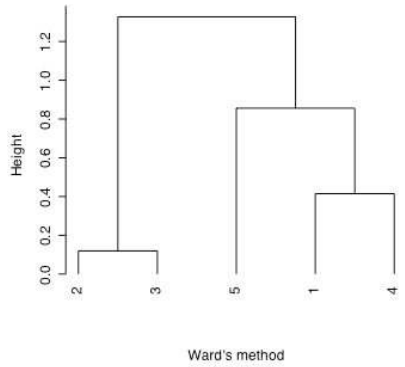
- $O(n^2)$ time and $O(n)$ space for all geometric methods.
- $O(n^2)$ time and $O(n^2)$ space for all graph (linkage) method.
- We have a parallel algorithm that *may* work well, with $O(n^2)$ time, in practice. But we can't be sure of this for an important criterion like the Ward minimum variance one.

Representation

Representing Hierarchical Trees: Motivation

1. For programming – implementation.
2. For interpreting or otherwise using the results of a hierarchical clustering.
3. To understand pitfalls and problems – inversions.
4. To see what betokens structure in one's data versus simply the way the hierarchy is displayed (artefact of display).
5. Uniqueness of the result – combinatorial properties.
6. Interesting linkages between dendrograms, oriented trees, permutations.
7. Different forms of output that we can expect, and possible use in areas like (structuring data to facilitate) information retrieval.

Representing Hierarchical Trees



- Enforcing a strict binary hierarchy is convenient. Hence: $n-1$ agglomerations, using n observations. (We will nearly always be concerned with strictly binary hierarchies.)
- Hierarchy is formed evidently from 5 observations. We actually have taken in this example a 5×3 matrix (of uniformly distributed values).
- We use the Ward mininimum variance criterion.
- The dendrogram is then given by: the sequence of agglomerations; the order of observations; and the heights of the merges.
- Different conventions can be used to represent the sequence of agglomerations.
- For n observations there are exactly $n-1$ agglomerations.

Representing Hierarchical Trees (commands in R)

```
> dat <- matrix(runif(15),nrow=5,ncol=3)
> dat
           [,1]      [,2]       [,3]
[1,] 0.380108950 0.1971235 0.09188467
[2,] 0.819335094 0.3892081 0.87644974
[3,] 0.805116058 0.4269140 0.76391744
[4,] 0.778452625 0.1690171 0.20353970
[5,] 0.007950265 0.3421287 0.56797563
> htemp <- hclust(dist(temp), method=''ward'')
> plclust(htemp, hang=-1, xlab=''Ward's method'', sub='' '')
> ht$merge
     [,1] [,2]
[1,]   -2   -3
[2,]   -1   -4
[3,]   -5    2
[4,]    1    3
> ht$height;  ht$order
[1] 0.1195301 0.4146499 0.8559081 1.3273986
[1] 2 3 5 1 4
```


Representing Hierarchical Trees: Avoiding Inversions

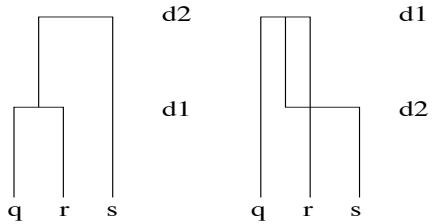
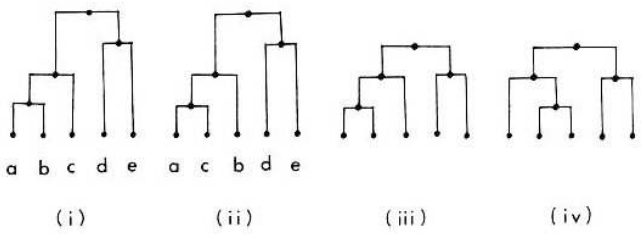


Figure: Alternative representations of a hierarchy with an inversion.

- Inversion impossible if: $d(i,j) < d(i,k)$ or $d(j,k) \Rightarrow d(i,j) < d(i \cup j, k)$
- Ability to agglomerate a pair of RNNs (reciprocal nearest neighbors) with no side effects on later agglomerations is the same as the reducibility property.
- Reducibility property (due to M. Bruynooghe, 1978, and used extensively in limited memory programs from the late 1970s onwards).

- Not satisfied by: centroid, median. Satisfied by: Ward, single link, complete link.
- Say a and b agglomerate into $c = a \cup b$. Consider some other cluster or object c'.
- $d(a,b) \leq \inf\{d(a,c'), d(bc')\} \Longrightarrow \{d(a,c'), d(b,c')\} \leq d(a \cup b, c')$
- If $d(a,b) \leq \rho \leq \inf\{d(a,c'), d(b,c')\}$ then $\rho \leq \inf\{d(a,c'), d(b,c')\} \leq d(c,c')$
- $d(a,b) \leq \rho$ and $d(a,c') \geq \rho$ and $d(b,c') \geq \rho \Longrightarrow d(c,c') \geq \rho$

Representing Hierarchical Trees: Isomorphisms



- In examining equivalent shape – isomorphisms – between dendrograms, we must distinguish between: (1) whether or not terminals are labeled, and (2) and whether or not we take into account ranks of agglomeration heights/levels.
- For binary dendrograms, we have: labeled, ranked (L-R); labeled, non-ranked

(L-NR); unlabeled, non-ranked (NL-NR); unlabeled, ranked (NL-R).

- In the Fig., (i) and (ii) are isomorphic NL-R dendgrams; but as L-R dendrograms they are not isomorphic.
- Considered as NL-NR dendrograms, all the dendrograms are isomorphic.

Table 1
Numbers of non-isomorphic dendrograms for four types of binary dendrogram

n	L-R $a(n)$	L-NR $b(n)$	NL-NR $c(n)$	NL-R $d(n)$
1	1	1	1	1
2	1	1	1	1
3	3	3	1	1
4	18	15	2	2
5	180	105	3	5
6	2700	945	6	16
7	56700	10395	11	61
8	1587600	135135	23	272
9	57153600	2027025	46	1385
10	2571912000	34459425	98	7936

Notes: n = number of terminal nodes, L = labelled, NL = unlabelled, R = ranked, NR = non-ranked.

Representing Hierarchical Trees: Sibson's Packed Form

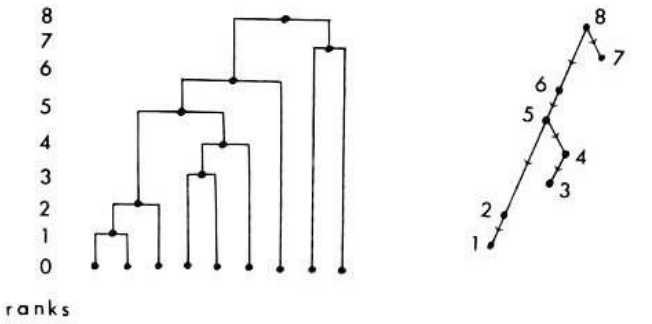


Fig. 3. Dendrogram ($n = 9$) and associated oriented binary tree.

- Sibson's (1973) packed representation or permutation representation of a dendrogram.
- (i) put lower ranked subtree always to the left; and (ii) read off oriented binary tree on non-terminal nodes.
- Then for any terminal node indexed by i, with the exception of the rightmost which will always be n, define $p(i)$ as the rank at which the terminal node is first united with some terminal node to its right.
- For the dendrogram shown, the packed representation is: $p = (125346879)$.
- This is also an *inorder* traversal of the oriented binary tree.
- The packed representation is a uniquely defined permutation of $1 \ldots n$.
- NL-R dendrograms (on n terminals) are isomorphic to either down-up permutations, or up-down permutations (both on $n - 1$ elements).
- Applications: (i) Sibson's $O(n^2)$ algorithm for the single link method; (ii) generating all possible dendrograms; (iii) generally, understanding what sort of beasts one is dealing with.

- Sibson (1973) and Defays (1977) described algorithms that updated a "packed" representation of the hierarchy for, resp., single and complete link. These were both $O(n^2)$ time and $O(n)$ space.
- References
  - F. Murtagh, "Counting dendrograms: a survey", Discrete Applied Mathematics, 7, 191–199, 1984
  - R. Sibson, "SLINK: an optimally efficient algorithm for the single-link cluster method", Computer Journal, 16, 30–34, 1973
  - On-Line Encyclopedia of Integer Sequences, www.research.att.com/∼njas/sequences/Seis.html
  - For more on single link, see FJ Rohlf, Single-link clustering algorithms, in *Handbook of Statistics, Vol. 2*, eds., PR Krishnaiah and LN Kanal, North-Holland, 267–284, 1982.

Representing Hierarchical Trees: Extreme Shapes

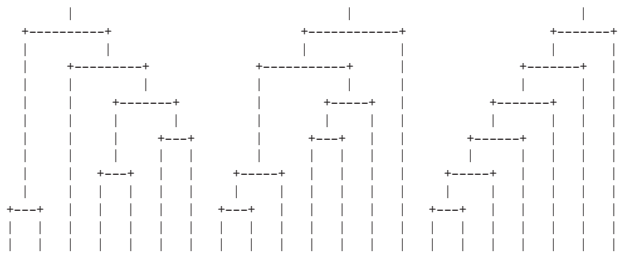


Figure: Three binary hierarchies: balanced, intermediate, and unbalanced, on $n = 7$ terminals.

- Consider n as an integer power of 2, e.g. $n = 8$. Agglomerate as: (12), (34), (56), (78), (1234), (5678), (12345678). This is what we want as the best

possible "balance" or "symmetry".

- By construction, the number of non-terminal nodes is always the same, viz. $n - 1$.
- The extreme "unbalanced" hierarchy has a path from root to terminals that is $n - 1$ long.
- Whereas the extreme "balanced" hierarchy has approximately equal path lengths $\log_2 n$ from root to terminals. (This path length is exactly logarithmic if n is an integer power of 2; and we consider our dendrograms as non-ranked. When looking at isomorphisms we referrd to such dendrograms as NL-NR or L-NR.)
- Application: tree traversal in information retrieval.
- Reference: F. Murtagh, "Structures of hierarchic clusterings: implications for information retrieval and for multivariate data analysis", Information Processing and Management, 20, 611-617, 1984

Normalizing Variables Prior to Clustering

- Correspondence analysis uses the χ^2 distance between rows and between columns
- The χ^2 distance is a weighted Euclidean distance between *profiles*
- So as input we have a set of objects, and a set of variables, with a χ^2 distance defined on each
- As output we have projections in a set of new axes (factors). In this space, the points (objects, variables) have a Euclidean distance defined on them
- This is very convenient... We take as input, say, frequency of occurence counts or ranks or various other forms of quantitative or qualitative data. We get as output constant weighted points endowed with the Euclidean distance.

Scores 5 students in 6 subjects

	CSc	CPg	CGr	CNw	DbM	SwE
A	54	55	31	36	46	40
B	35	56	20	20	49	45
C	47	73	39	30	48	57
D	54	72	33	42	57	21
E	18	24	11	14	19	7

	CSc	CPg	CGr	CNw	DbM	SwE
mean profile:	.18	.24	.12	.12	.19	.15
profile of D:	.19	.26	.12	.15	.20	.08
profile of E:	.19	.26	.12	.15	.20	.08

Scores (out of 100) of 5 students, A–E, in 6 subjects. Subjects: CSc: Computer Science Proficiency, CPg: Computer Programming, CGr: Computer Graphics, CNw: Computer Networks, DbM: Database Management, SwE: Software Engineering.

Scores 5 students in 6 subjects (Cont'd.)

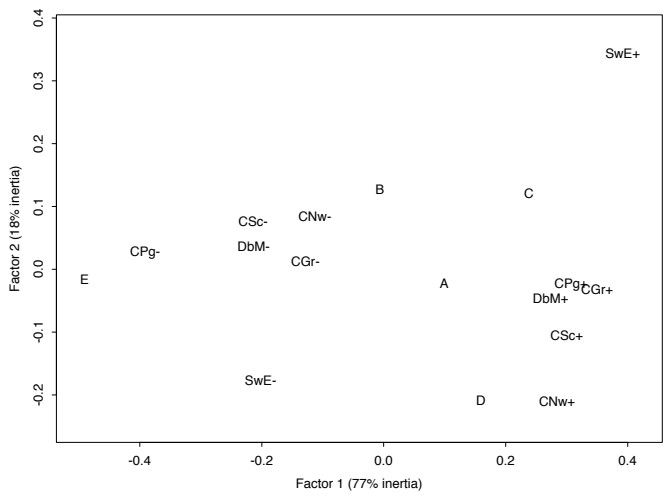
- Correspondence analysis highlights the similarities and the differences in the profiles.
- Note that all the scores of D and E are in the same proportion (E's scores are one-third those of D).
- Note also that E has the lowest scores both in absolute and relative terms in all the subjects.
- D and E have identical profiles: without data coding they would be located at the same location in the output display.
- Both D and E show a positive association with CNw (computer networks) and a negative association with SwE (software engineering) because in comparison with the mean profile, D and E have, in their profile, a relatively larger component of CNw and a relatively smaller component of SwE.

- We need to clearly differentiate between the profiles of D and E, which we do by *doubling* the data.
- Doubling: we attribute two scores per subject instead of a single score. The "score awarded", $k(i, j^+)$, is equal to the initial score. The "score not awarded", $k(i, j^-)$, is equal to its complement, i.e., $100 - k(i, j^+)$.
- Lever principle: a "$+$" variable and its corresponding "$-$" variable lie on the opposite sides of the origin and collinear with it.
- And: if the mass of the profile of j^+ is greater than the mass of the profile of j^- (which means that the average score for the subject j was greater than 50 out of 100), the point j^+ is closer to th e origin than j^-.
- We will find that except in CPg, the average score of the students was below 50 in all the subjects.

Data coding: Doubling

	CSc+	CSc-	CPg+	CPg-	CGr+	CGr-	CNw+	CNw-	DbM+	DbM-	SwE+	SwE-
A	54	46	55	45	31	69	36	64	46	54	40	60
B	35	65	56	44	20	80	20	80	49	51	45	55
C	47	53	73	27	39	61	30	70	48	52	57	43
D	54	46	72	28	33	67	42	58	57	43	21	79
E	18	82	24	76	11	89	14	86	19	81	7	93

Doubled table of scores derived from previous table. Note: all rows now have the same total.



χ^2 Distance on Input Data, Euclidean Distance on Output Factors

- *Principle of distributional equivalence:* Consider two elements j_1 and j_2 of J with identical profiles: i.e. $f_I^{j_1} = f_I^{j_2}$. Consider now that elements (or columns) j_1 and j_2 are replaced with a new element j_s such that the new coordinates are aggregated profiles, $f_{ij_s} = f_{ij_1} + f_{ij_2}$, and the new masses are similarly aggregated: $f_{ij_s} = f_{ij_1} + f_{ij_2}$. Then there is *no effect* on the distribution of distances between elements of I. The distance between elements of J, other than j_1 and j_2 is naturally not modified.
- The principle of distributional equivalence leads to representational self-similarity: aggregation of rows or columns, as defined above, leads to the same analysis. Therefore it is very appropriate to analyze a contingency table with fine granularity, and seek in the analysis to merge rows or columns, through aggregation.
- Have that the χ^2 metric is defined in direct space, i.e. space of profiles.

- The Euclidean metric is defined for the factors.
- We can characterize correspondence analysis as the mapping of a cloud in χ^2 space to Euclidean space.
- So weighting of observations and of variables is carried out in an "integral" or "inbuilt" way in Correspondence Analysis.
- And the output representation is unweighted Euclidean.
- *This is a very convenient way therefore to handle weighting of input data... carry out Correspondence Analysis first, and input projections on the most important factors to the hierarchical clustering.*

Graph Methods

- MST, minimal spanning tree, and its relationship with single linkage hierarchical clustering
- Other graph structures – Voronoi diagram and its dual, the Delaunay triangulation
- Clustering *on* graphs, implying
  - Graph defines a contiguity constraint
  - Contiguity-constrained single linkage
  - Contiguity-constrained complete linkage

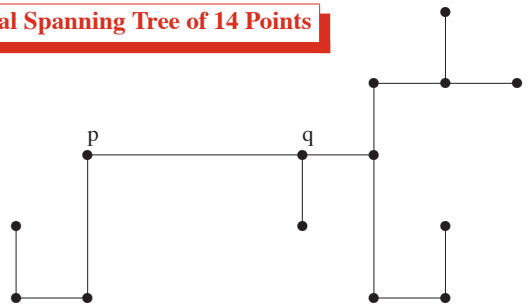
Minimal Spanning Tree

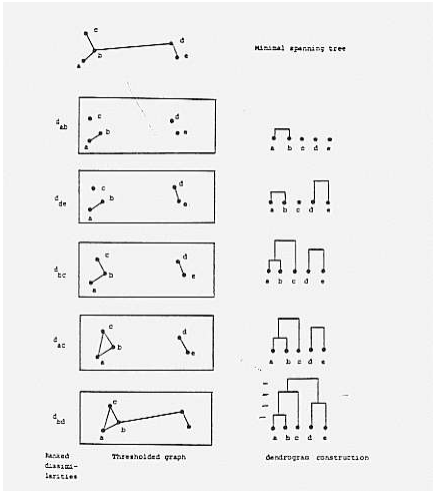
Step 1 Select an arbitrary point and connect it to the least dissimilar neighbor. These two points constitute a subgraph of the MST.

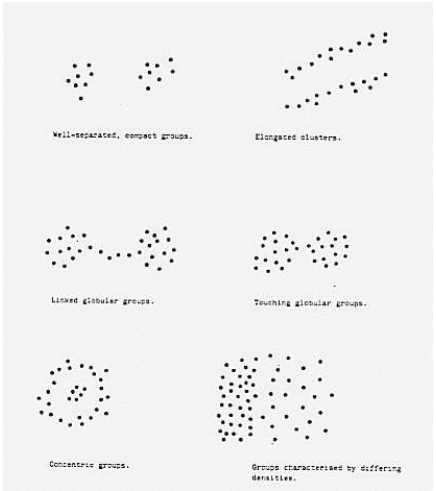
Step 2 Connect the current subgraph to the least dissimilar neighbor of any of the members of the subgraph.

Step 3 Loop on Step 2, until all points are in the one subgraph: this, then, is the MST.

Minimal Spanning Tree of 14 Points



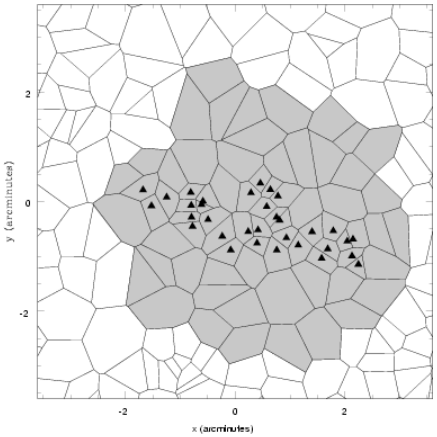




Voronoi Diagram

- M. Ramella, W. Boschin, D. Fadda and M. Nonino, Finding galaxy clusters using Voronoi tessellations, A&A 368, 776-786 (2001)
- For lots on Voronoi diagrams: http://www.voronoi.com/cgi-bin/display.voronoi_applications.php?cat=Applications
- Voronoi diagram: for given points i, we define the Voronoi cell or region of i as $\{x|d(x,i) \leq d(x,i')\}\ \ \forall i'$.
- Delaunay triangulation: perpendicular bisectors of Voronoi boundaries.
- Demo: http://www.csie.ntu.edu.tw/∼b5506061/voronoi/Voronoi.html
- Theorem: MST $\subset$ Delaunay triangulation.

Voronoi Diagram



Some galaxies are shown here.

Efficiency of Graph Clustering Algorithms

- For MST, the Prim-Dijkstra and Kruskal algorithms, and usually the Sollin parallel one, are to be found in *every* textbook on computer algorithms.
- NN-chains and RNNs can be used too. If NNs can be found quickly then this can be of great advantage.
- For sparse graphs, the number of edges may be $<< O(n^2)$. Such a case is when the graph is planar (for $m > 1, m \leq 3n - 6$: see any book on graph theory).
- Then $O(m \log n)$ algorithms can be easily found, where m is the number of edges.

Contiguity-Constrained Clustering Algorithms

- There are many ways of formulating this problem.
- Consider the case of a graph that expresses a constraint: only vertices that have an edge linking them can be clustered.
- *Contiguity-constrained single linkage clustering*: At each agglomeration, fuse together the two clusters of least interconnecting dissimilarity, such that this dissimilarity is between a pair of contiguous objects.
- We are simultaneously constructing the MST of the contiguity graph. We can take it and transform to a single link hierarchy.
- The contiguity-constrained single link method cannot give rise to an inversion.
- Computational complexity: $O(m \log n)$ for m edges and n vertices.
- *Contiguity-constrained complete linkage clustering*: We allow agglomeration – using any of the usual criteria – such that there exists a contiguity link between

at least one member of each of the clusters.

- Of the major hierarchical methods, only the complete link method excludes the possibility of inversions.

  For proof, see Murtagh, Multidimensional Clustering Algorithms, Physica-Verlag, 1985 – scanned on the CSNA Service CD, chapter 5, section 2, pp. 124–126.
- Computational complexity: $O(n^2)$ time and $O(n^2)$ space, by checking for a contiguity link in the context of the NN-chain algorithm.

Other Clustering Paradigms

- All three "clustering paradigms" – families of methods – are based on the EM, expectation-maximization, criterion optimization algorithm.
- First, partitioning.
- Second, Gaussian mixture modeling.
- Third, Kohonen self-organizing feature map.

AP Dempster, NM Laird and DB Rubin, "Maximum likelihood from incomplete data via the EM algorithm", Journal of the Royal Statistical Society, B, 39, 1-38, 1977

Partitioning

Iterative optimization algorithm for the variance criterion

Step 1 Arbitrarily define a set of k cluster centres.

Step 2 – M-step Assign each object to the cluster to which it is closest (using the Euclidean distance, $d^2(i,p) = \|\mathbf{i} - \mathbf{p}\|^2$).

Step 3 – E-step Redefine cluster centres on the basis of the current cluster memberships.

Step 4 If the totalled within class variances is better than at the previous iteration, then return to Step 2.

Partitioning – Properties

- Sub-optimal.
- Dependent on initial cluster centres.
- The two main steps define the EM algorithm. Expectation = mean; and Maximization = assignment step.
- Many other algorithms are similar. For instance, Edwin Diday's *nuées dynamiques* (dynamical clouds) method ran k-means lots of times and took the consensus result.
- Widely used (since it is fast, whereas computational cost of hierarchical clustering is usually $O(n^2)$).

Partitioning: Späth's Exchange Algorithm

Exchange method for the minimum variance criterion

Step 1 Arbitrarily choose an initial partition.

Step 2 For each $i \in p$, see if the criterion is bettered by relocating i in another class q. If this is the case, we choose class q such that the criterion V is least; if it is not the case, we proceed to the next i.

Step 3 If the maximum possible number of iterations has not been reached, and if at least one relocation took place in Step 2, return again to Step 2.

H. Späth, Cluster Dissection and Analysis, Ellis Horwood, 1985

Exchange Algorithm – Properties

- Clusters will not become empty.
- The change in variance brought about by relocating object i from class p to class q can be shown to be $\frac{|p|}{|p|-1}\|\mathbf{i} - \mathbf{p}\|^2 - \frac{|q|}{|q|-1}\|\mathbf{i} - \mathbf{q}\|^2$

Mixture Modeling

- Data is a mixture of G multivariate Gaussians:

$$f_k(x;\theta) \sim \text{MVN}(\mu_k, \Sigma_k) \qquad k = 1, \ldots, G$$

$$f(x;\theta) = \sum_{k=1}^{G} \pi_k f_k(x;\theta)$$

Mixing or prior probabilities, $\sum_{k=1}^{G} \pi_k = 1$

- Estimate parameters θ, π by maximizing the mixture likelihood:

$$L(\theta, \gamma) = \Pi_{i=1}^{n} f(x_i;\theta)$$

where x_i is the ith observation, and γ is a cluster assignment function.

Mixture Modeling – 2

- Implementation: hierarchical agglomerative; iterative relocation; EM; start with agglomerative and refine with EM.
- Choosing the number of clusters – the Bayes Information Criterion (BIC).
  Bayes factor, $B = p(x \mid M_2)/p(x \mid M_1)$
  $p(x \mid M_2)$ = integrated likelihood of the mixture model 2 obtained by integrating over parameter space.
- Approximate the Bayes factor by the BIC:
  Let $p(x \mid G)$ be the integrated likelihood of the data given that there are G clusters.
  Then:
  $2 \log p(x \mid G) \approx 2l(x;\hat{\theta}, G) - m_G \log n = BIC$
  $l(x;\hat{\theta}, G)$ is the maximized mixture log-likelihood with G clusters.

m_G is the number of independent parameters to be estimated in the G-cluster model.

The larger the value of BIC, the better the model.

Example: Gamma-Ray Bursts

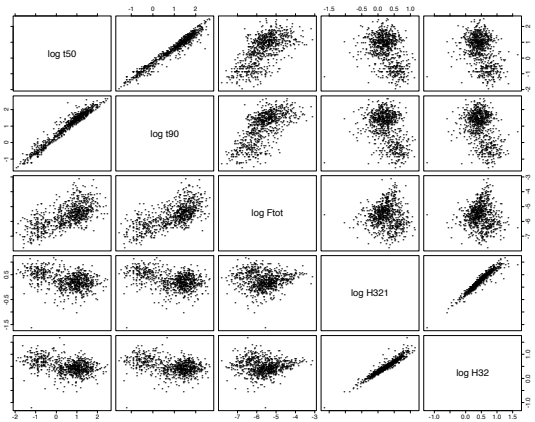
- Few gamma-ray burst (GRB) sources have astronomical counterparts at other wavebands. Hence empirical studies of GRBs have been largely restricted to the analysis of their gamma ray properties.
- Bulk properties such as fluence and spectral hardness are used.
- Studies fall into two categories: examination whether GRB bulk properties comprise a homogeneous population or are divided into distinct classes; and search for relationships between bulk properties.
- Generally accepted taxonomy of GRBs is division between short-hard and long-soft bursts.
- We use GRBs from the Third BATSE Catalog, from the Compton Gamma Ray Observatory. Data from 1996.
- There are roughly eleven variables of potential astrophysical interest: two

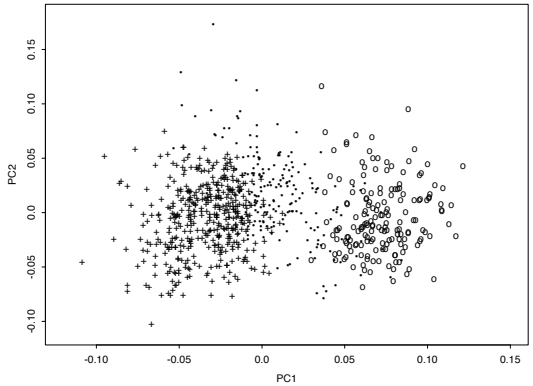
measures of location in Galactic coordinates, l and b; two measures of burst durations, the times within which 50% (T_{50}) and 90% (T_{90}) of the flux arrives; three peak fluxes P_{64}, P_{256} and P_{1024} measured in 64 ms, 256 ms and 1024 ms bins respectively; and four time-integrated fluences $F_1 - F_4$ in the 0-50 keV, 50-100 keV, 100-300 keV and $>$ 300 keV spectral channels respectively

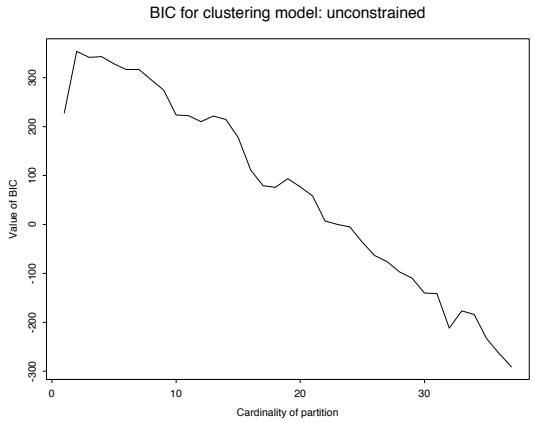
- Consider three composite variables: the total fluence, $F_T = F_1 + F_2 + F_3 + F_4$, and two measures of spectral hardness derived from the ratios of channel fluences, $H_{32} = F_3/F_2$ and $H_{321} = F_3/(F_1 + F_2)$. Of the 1122 listed bursts, 807 have data on all the variables described above.
- Our sample had 797 GRBs. For some analyses, we also used a subset of 644 bursts with 'debiased' durations, T_{90}^d. Here the durations are modified to account for the effect that brighter bursts will have signal above the noise for longer periods than fainter bursts with the same time profiles.
- We use log variables, rather than normalized or standardized variables.
- Our analysis was performed using log T_{50}, log T_{90}, log F_{tot}, log P_{256}, log H_{321} and log H_{32}.

Example: Gamma-Ray Bursts. Plots To Follow.

- Reference: S. Mukherjee, E.D. Feigelson, G.J. Babu, F. Murtagh, C. Fraley and A. Raftery, "Three types of gamma ray bursts", The Astrophysical Journal, 508, 314-327, 1998.
- Pairwise plots of BATSE data showing strong correlation between variables 1 and 2, and 4 and 5.
- 3-cluster results on unconstrained model clustering (on variables 1, 3 and 4) in principal component space.
- Corresponding BIC values with maximum value corresponding to the 3-cluster solution.







Raftery's Cluster Modeling

- We will parametrize the standard spectral decomposition of Σ_k:
  $\Sigma_k = \lambda_k D_k A_k D_k^T$
  λ_k is largest eigenvalue of Σ:
  controls volume of cluster.
  D_k is matrix of eigenvectors:
  controls orientation of cluster.
  A_k is diag$\{1, \alpha_{2k} \ldots \alpha_{pk}\}$:
  controls shape of cluster.
- Example 1: set shape, different sizes and orientations:
  For $p = 2$ dimensional data,
  $A_k = \text{diag}\{1, \alpha\}, \alpha = \lambda_2/\lambda_1$
  $\alpha < 1 \Longrightarrow$ long and narrow cluster.
  Use: finding aligned sets of points.

Raftery's Cluster Modeling – 2

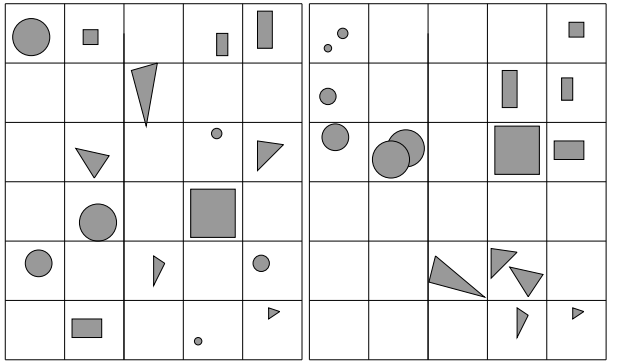
- Example 2: hyperspherical clusters, different sizes: $\Sigma_k = \lambda_k I$ (I = identity matrix).
- Example 3: hyperspherical, same size (Ward's method): $\Sigma_k = \lambda I$.
- Example 4: unconstrained Σ_k.
  A.J. Scott and M.J. Symons, "Clustering methods based on likelihood ratio criteria", Biometrics, 27, 387–397, 1971.
  W_k = SSCP matrix for cluster k,
  x_k = mean of cluster k,
  n_k = cardinality of cluster k,
  $W_k = \sum_{i \in \text{ cluster}} (x_i - x_k)(x_i - x_k)^T$
  W_k/n_k = MLE of Σ_k.
  Maximize $\sum_{k=1}^{G} n_k \log \left| \frac{W_k}{n_k} \right|$ (| . | = det).

Kohonen Self-Organizing Feature Map

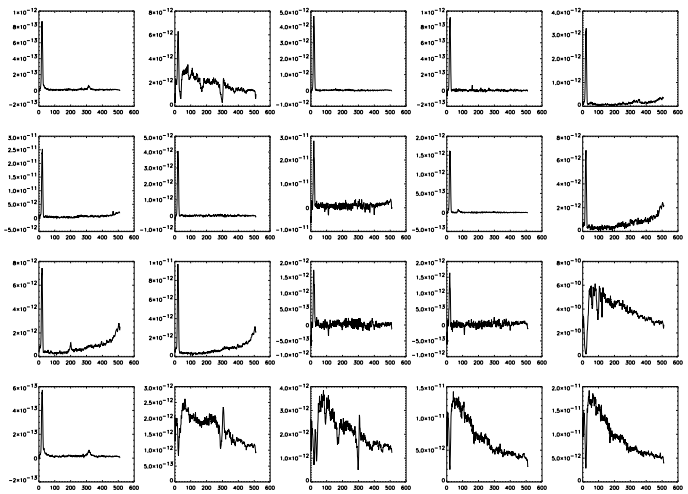
- Regular grid output representational or display space.
- Determine vectors w_k, such that inputs x_i are parsimoniously summarized (clustering objective); and in addition the vectors w_k are positioned in representational space so that similar vectors are close (low-dimensional projection objective) in *representation space*.
- **Clustering:** Associate each x_i with some one w_k such that
  $k = argmin \parallel x_i - w_k \parallel$
  Low-Dimensional projection:
  $\parallel w_k - w_k' \parallel < \parallel w_k - w_k'' \parallel \Longrightarrow \parallel k - k' \parallel \leq \parallel k - k'' \parallel$
- Initial random choice of values for w_k.
- Update the set of w_k ($\forall k$) on the basis of presentation of input vectors, x_i.
- Processing one x_i is termed an iteration. Going through all x_i once is termed an

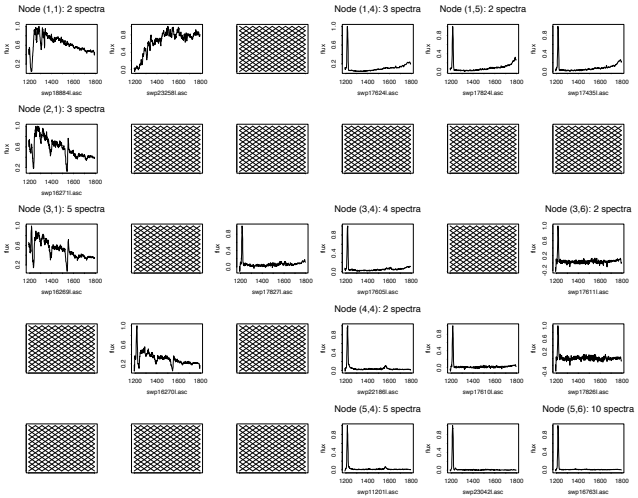
epoch.

- Update not just the so-called winner w_k, but also neighbors of w_k with respect to the representational space.
- The neighborhood is initially chosen to be quite large (e.g. a 4×4 zone) and as the epochs proceed, is reduced to 1×1 (i.e. no neighborhood).
- Example: set of 45 spectra of the complex AGN (active galactic nucleus) object, NGC 4151, taken with the IUE (International Ultraviolet Explorer) satellite.
- 45 spectra observed with the SWP spectral camera, with wavelengths from 1191.2 Å to approximately 1794.4 Å, with values at 512 interval steps.
- We will show sample of 20 spectra; and then Kohonen map of these.
- Murtagh, F. and M. Hernández-Pajares, "The Kohonen self-organizing map method: an assessment", Journal of Classification, 12, 165–190, 1995



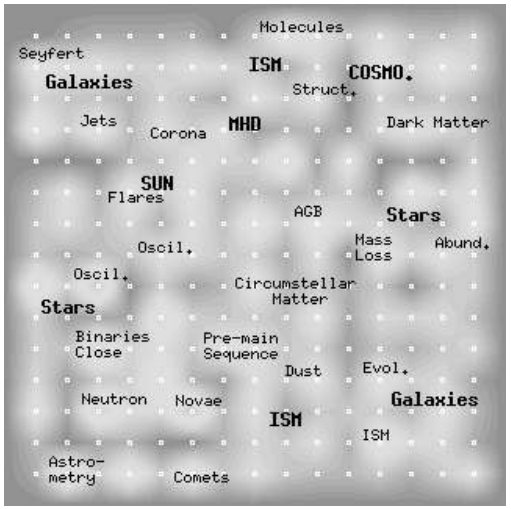
Left: input; right: Kohonen output.

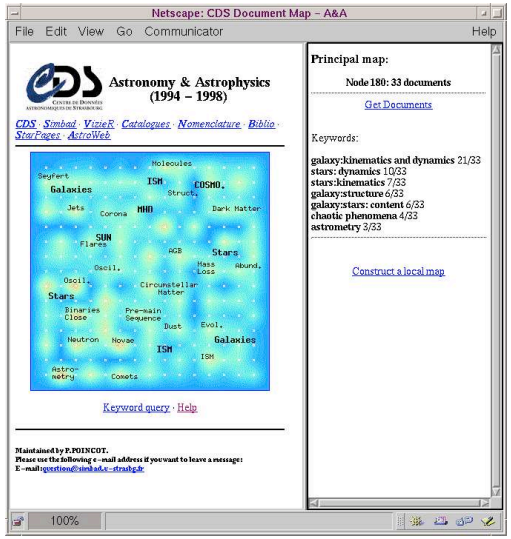


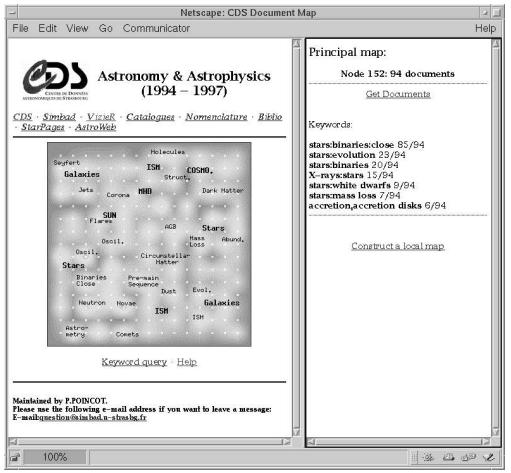


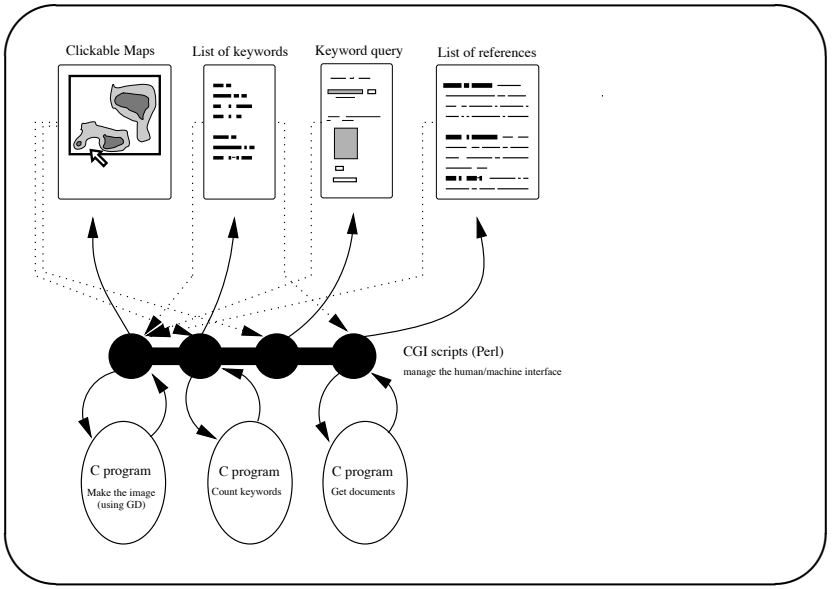
Kohonen Map: Interactive User Interface

- About 10,000 documents described by 269 keywords from articles published in A&A; also in ApJ.
- 15×15 grid was used for the principal map, and a 5×5 grid for detailed maps.
- User clicks on thematic area, or enters keywords.
- A detailed map is produced. Any document listed allows access to the full document through ADS.
- This system is server-side, based on imagemap and CGI scripts.









Software

Software: http://astro.u-strasbg.fr/∼fmurtagh/mda-sw

- Hierarchical clustering, Ward minimum variance criterion, supporting weighting of observations.
- In C, and in R. Also in Java.
- With correspondence analysis programs in R. And in Java.