Index of Lecture 9: Multivariate distance and (first) applications

Practical Information

Today's lecture — classical statistical material on multivariate distances and two approaches to visualizing distances,

* multi-dimensional scaling (MDS; Manly 3/4, Chapter 11[1]; ED, Chapter 5),
* cluster analysis (CA) with main focus on hierarchical clustering (Manly 3/4, Chapter 9)[2] and a brief introduction to K-means partitioning (SL, Chapter 10),

— the purpose of analysis is descriptive/explorative, and for most parts without any attempt at statistical inference.

Links with later ideas/material:

* MDS related to dimension-reduction approaches, conceptually and practically,
* CA is one example of unsupervised learning, but is often applied to data with some knowledge of structures/groups, and can then be viewed as a descriptive supplement to supervised learning methods based on that knowledge.

Other news:

- home assignment #4 due today,
- project proposal due next Thursday.

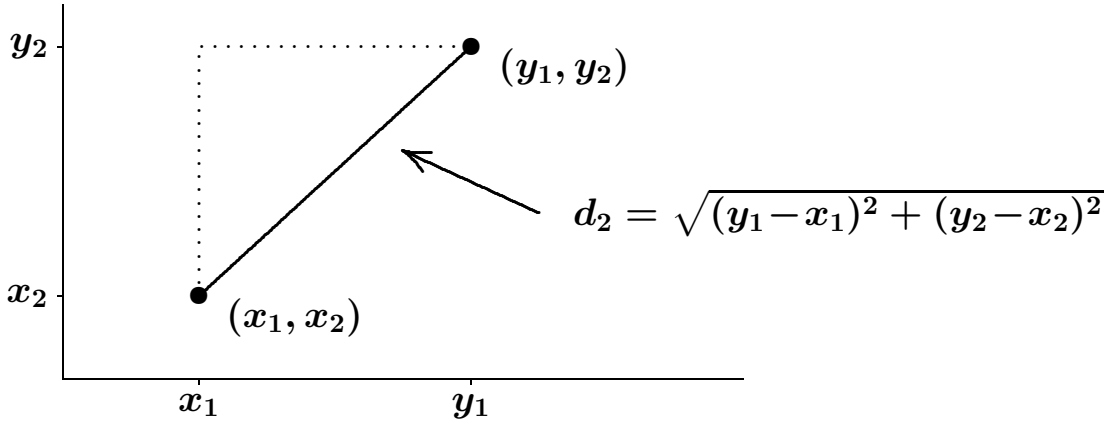
[1] We postpone a section in Manly 3/4, Chapter 12 on principal coordinates analysis, a particular version of MDS.
[2] Also covered in ED, Chapter 6; SL, Chapter 10.

Euclidean and Other Distances

Euclidean (sometimes also L_2 or L2) distance:

- the "straight-line" distance between two points (in dimensions 1, 2 and 3), and generalized to higher dimensions
- has a simple computational formula (due to ancient Greek mathematicians): $d_2(\mathrm{x},\mathrm{y}) = \sqrt{\sum_{j=1}^{p}(x_j - y_j)^2}$,

  for points $\mathrm{x} = (x_1, \ldots, x_p)$ and $\mathrm{y} = (y_1, \ldots, y_p)$ in p-dimensional space,
- simple illustration in 2-dimensional space:



- alternative distances (norms):
  - L_1 ("city block"): $d_1(\mathrm{x},\mathrm{y}) = \sum_{j=1}^{p} |x_j - y_j|$,
  - L_∞ (supremum): $d_\infty(\mathrm{x},\mathrm{y}) = \max_{j=1,\ldots,p} |x_j - y_j|$,

  all (including d_2) assuming quantitative (continuous) values,
- (terminology) distance $\sim$ dissimilarity; distance ≈ 0 for close points.

Euclidian Distance: Dogs

Summary: 6 lower jaw measurements on craniums of prehistoric dogs in Thailand and 6 other possibly related species; interest is in quantifying distances between species.

First step: standardize the measurements/variables by subtracting their mean and dividing by their standard deviations?

* to give variables same weight in distances (but makes interpretation more difficult),
* variable ratios between standard deviations (range ≈ 1–5) $\Rightarrow$ best to standardize.

Second step: calculate d_2 between all pairs of observations (species), and collect in a distance matrix:[3]

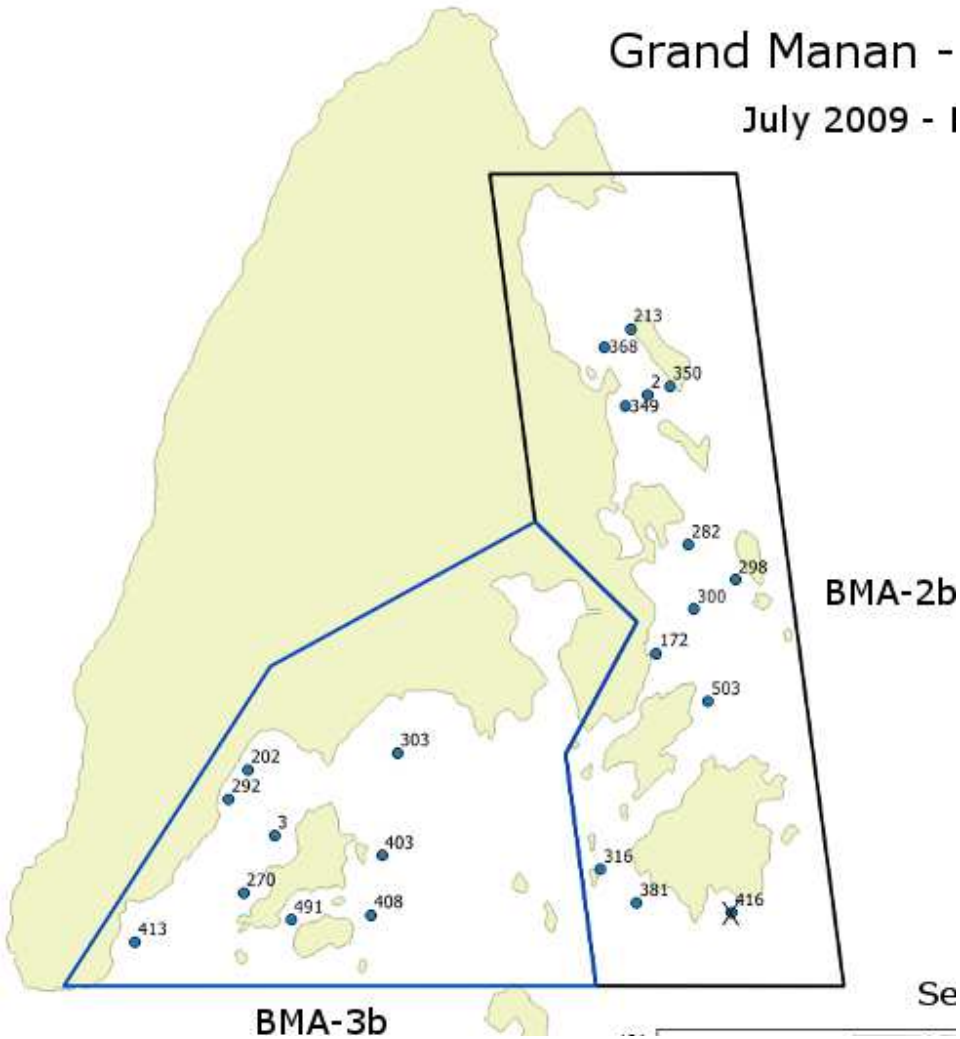
Species	Species number 1	2	3	4	5	6	7
1. Modern dog	–						
2. Golden jackal	1.91	–					
3. Chinese wolf	5.38	7.11	–				
4. Indian wolf	3.38	5.06	2.14	–			
5. Cuon	1.51	3.19	4.57	2.91	–		
6. Dingo	1.56	3.18	4.21	2.20	1.67	–	
7. Prehistoric dog	0.66	2.39	5.12	3.24	1.26	1.71	–

* in the distance matrix, diagonal could be filled with 0s, and upper triangle could be filled as well (by symmetry),
* interpretation: Modern dog is clearly "closest" relative to Prehistoric dog (0.66); same conclusion is reached without standardization.

[3] Minitab: distance matrix from `Multivariate-Cluster Obs.-Storage` menu.

Example of 2-D Distances: Salmon Farms

To illustrate distance concepts,
we consider Euclidean and sea-way[4]
distances for salmon farms
at Grand Manan, NB:



[4] Sea-way distance: the shortest distance between two points (at sea) when travelling by sea instead of by air; for details, see e.g. Cameron (2016): `https://www.ausvet.com.au/seaway-distances-with-postgresql/`.

Distance Matrices for GM Farms

Euclidean distances (km) for BMA 3b:

Farm	413	270	491	408	3	403	292	202	303
413	0	–	–	–	–	–	–	–	–
270	2.74	0	–	–	–	–	–	–	–
491	3.63	1.26	0	–	–	–	–	–	–
408	5.48	2.99	1.85	0	–	–	–	–	–
3	4.05	1.51	1.98	2.88	0	–	–	–	–
403	6.03	3.31	2.56	1.39	2.51	0	–	–	–
292	3.94	2.19	3.13	4.23	1.35	3.76	0	–	–
202	4.75	2.84	3.59	4.38	1.63	3.66	0.81	0	–
303	7.45	4.80	4.56	3.78	3.40	2.39	4.03	3.46	0

Sea-way distances (km) for BMA 3b:

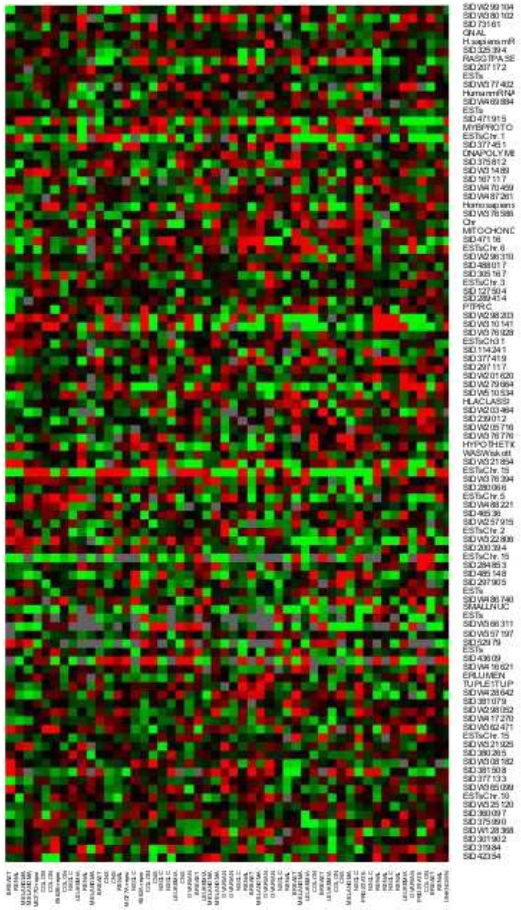
Farm	413	270	491	408	3	403	292	202	303
413	0	–	–	–	–	–	–	–	–
270	2.94	0	–	–	–	–	–	–	–
491	3.81	2.08	0	–	–	–	–	–	–
408	5.69	3.95	1.87	0	–	–	–	–	–
3	4.21	1.62	3.70	4.59	0	–	–	–	–
403	6.51	4.77	2.69	1.50	3.29	0	–	–	–
292	4.17	2.29	4.27	5.35	1.39	4.05	0	–	–
202	5.04	2.89	4.97	5.19	1.77	3.89	0.86	0	–
303	7.84	5.16	5.11	4.01	3.63	2.51	4.33	3.60	0

- some distances almost same, e.g.: $(\mathbf{3}, \mathbf{292})$, $(\mathbf{202}, \mathbf{270})$,
- some distance quite different, e.g.: $(\mathbf{3}, \mathbf{491})$, $(\mathbf{292}, \mathbf{408})$.

Distance in High Dimension: NCIARRAY Data

Microarray data for 64 (human) cancer cell lines of the National Cancer Institute,[5]

- gene expressions (light intensity): quantitative, typical range $(-6, 6)$,
- a total of 6830 genes, so high multivariate dimension,
- display of a heat map for 100 genes (Figure 1.3 of ESL):



Euclidean distances between samples: computed from 6830 squared deviations between gene expressions, e.g.

Sample	1 (CNS)	2 (CNS)	3 (CNS)	4 (REN)	5 (BRE)
1 (CNS)	0	–	–	–	–
2 (CNS)	51.4	0	–	–	–
3 (CNS)	65.9	69.0	0	–	–
4 (REN)	80.0	81.7	71.7	0	–
5 (BRE)	92.7	95.8	79.0	78.9	0

[5] Data from (ESL) Hastie, Tibshirani & Friedman (2009), *The Elements of Statistical Learning*, 2nd ed.; also (SL) Section 10.6.

Multidimensional Scaling: Idea and First Steps

Objective of MDS — may be stated as to:

- seek a configuration in d-dimensional space such that distances between points best match a distance matrix,
- construct a diagram showing the relationships between a number of objects, given only a distance matrix between the objects,
- visualize the distances/levels of similarity of individual cases of a dataset,
  - with high-dimensional data, we cannot visualize all dimensions, so try to represent a few, most important dimensions ($\sim$ dimension-reduction!) in a map.

Inputs for MDS may be:

i) distance (dissimilarity) matrix between observations, or

ii) variables measured for all observations from which distances are to be computed, e.g. using Euclidean distance.

Versions/Algorithms of MDS:

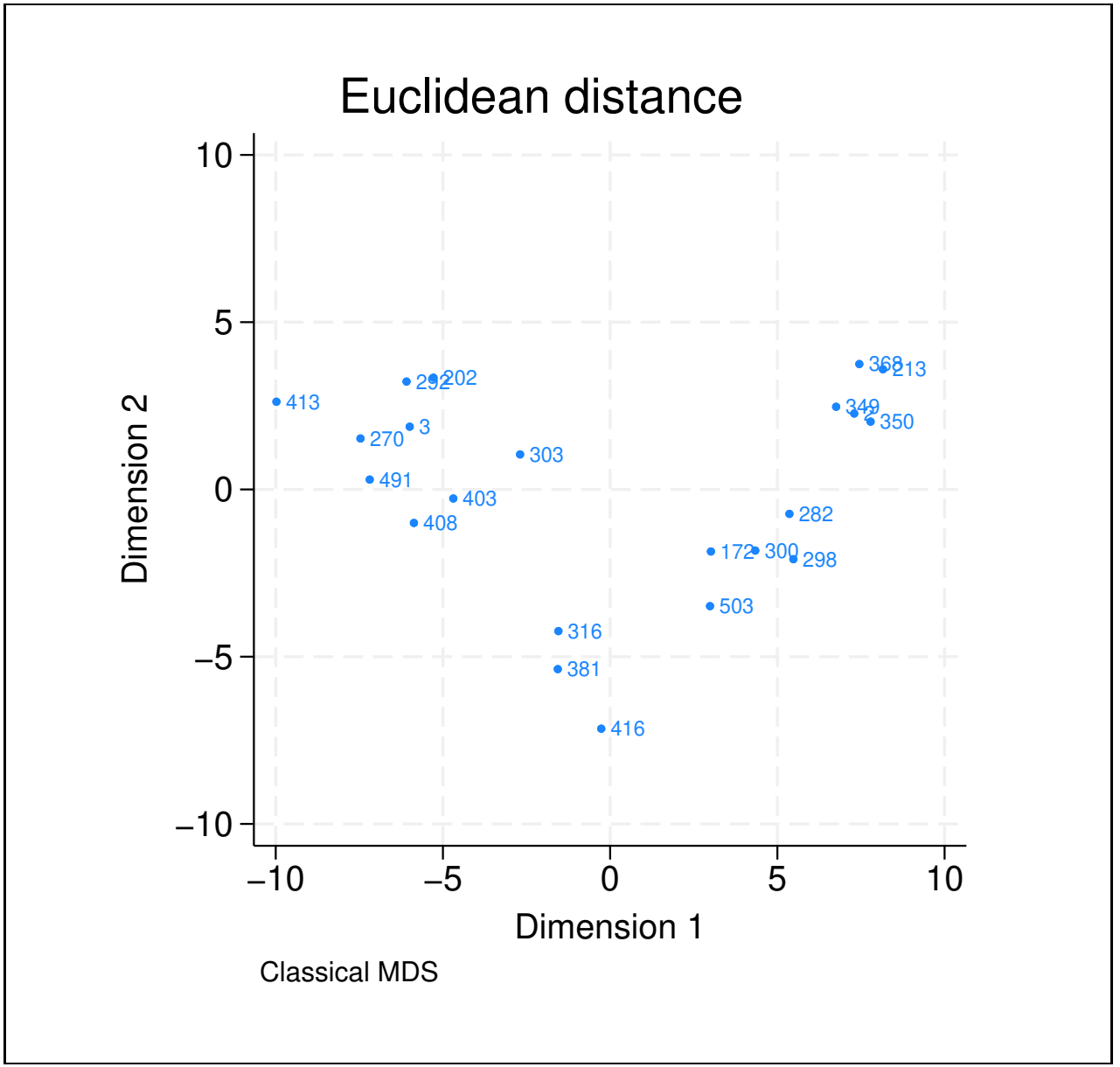
1) "classical" MDS: translates the problem into a matrix analysis problem with eigenvalues and eigenvectors $\Rightarrow$ some links with methods covered later in course,

2) "modern" MDS: involves iterative optimization a particular function over possible configurations; includes the classical method as a special case.

MDS for Euclidean Distance in GM

Should reproduce the real locations exactly!

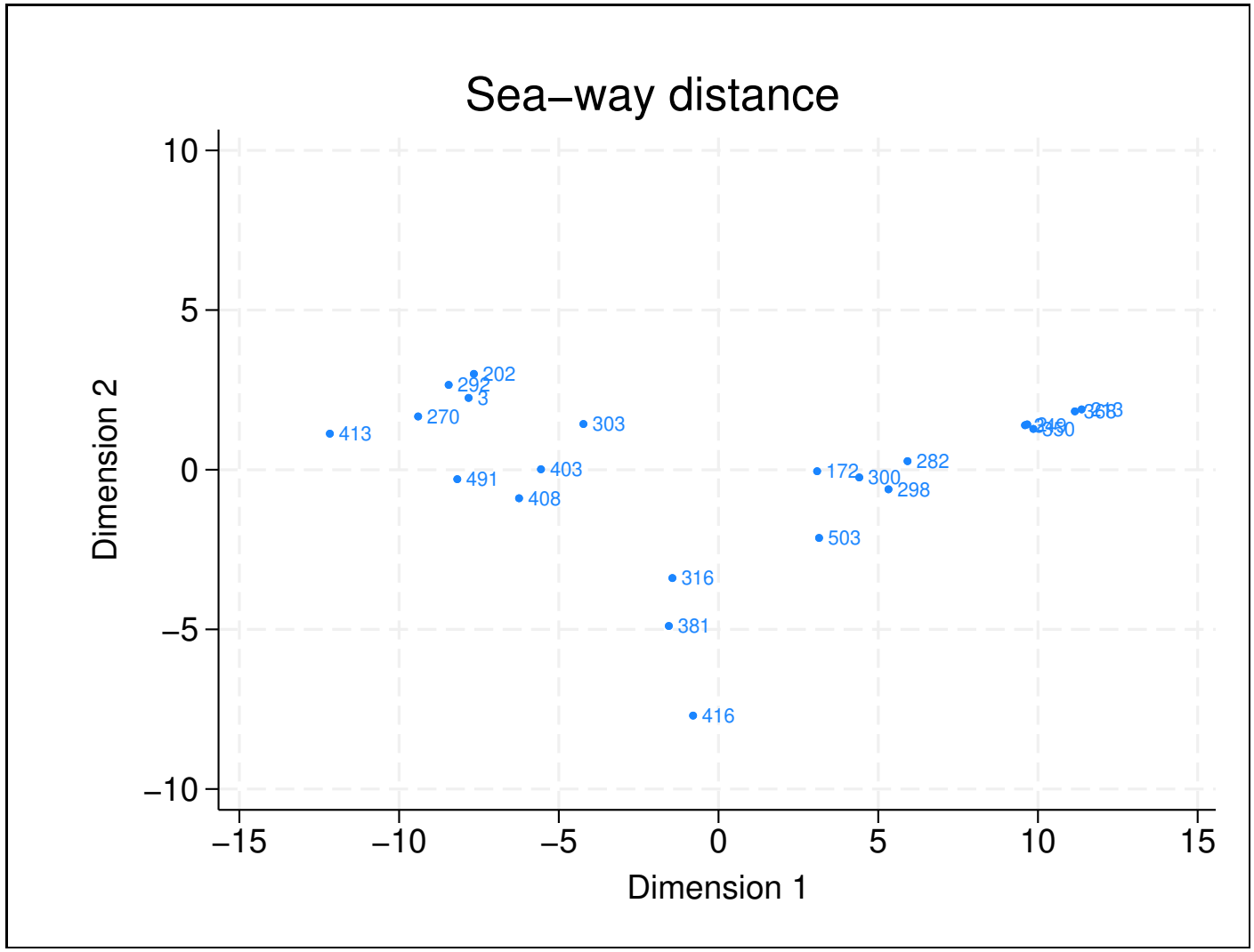
- up to scaling and rotation (incl. mirroring),
- 100% (lossless) representation.



MDS for Sea-way Distance in GM

Should not reproduce the real coordinates (i.e., some loss in representation) because of non-Euclidean distances,

- best visual representation of distances possible,
- appears to stretch points more in horizontal than vertical dimension.



Classical MDS Algorithms

First idea: translate the problem into a matrix analysis framework for a suitably defined matrix B (details in Manly or ED) whose eigenvalues/eigenvectors are computed,

* issue with matrix B: may not be positive definitive for non-Euclidean distance matrix,
* for Euclidean distance, also called principal coordinates analysis, and is equivalent to principal components analysis (next week) for the covariance matrix,
* also corresponds to modern MDS with a specific goodness-of-fit function (below).

Second idea: a representation of fit is achieved by "adapting" the distances between observations (δ_{ij}) to the MDS point configuration distances (d_{ij}, also dissimilarities), e.g. by a simple linear regression,

$$d_{ij} = \beta_0 + \beta_1 \cdot \delta_{ij} + \varepsilon_{ij}, \quad 1 \le i < j \le n, \tag{1}$$

in which case the fitted values, ($\hat{d}_{ij}$), minimize Kruskal's stress[6] measure,

$$(\text{"Stress 1"})^2 = \textstyle\sum_{ij}(d_{ij} - \hat{d}_{ij})^2 \;/\; \sum_{ij} d_{ij}^2 \,. \tag{2}$$

* ($\hat{d}_{ij}$) are the adjusted/fitted data distances (also, disparities),
* "Stress 1" measures goodness-of-fit, and small values (close to 0) are preferable; e.g. one may make decisions about reducing the number of dimensions on the resulting "Stress 1" value.

[6] "The word stress is used here because the statistic is a measure of the extent the spatial configuration of points has to be stressed in order to obtain the data distances d_{ij}." (Manly); note that the formula can be scaled by both dissimilarities and disparities (`stress` and `nstress` in Stata, respectively).

Modern MDS Algorithms

Why is there a need to improve/extend the method? — may be difficult to get good low-dimensional representations of large datasets (n large).

New idea: seek optimal MDS configuration of points (in d-dimensional space) to minimize goodness-of-fit criterion, for larger classes of solutions:

- metric scaling: extend linear relation in Eq. (1) to power or polynomial relations,
- nonmetric scaling: extend Eq. (1) to general monotonic functions $f : \delta_{ij} \mapsto f(\delta_{ij})$,
- other goodness-of-fit functions may work better with generalizations of Eq. (1).

Outline of modern algorithmic approach (following Manly):

1) set up a start configuration $\mathcal{C}$ for n objects in d-dimensional space (d fixed),
2) compute the Euclidean distances $d_{ij} = d_{ij}(\mathcal{C})$,
3) regress by Eq. (1), or extensions hereof, the configuration distances d_{ij} on the true distances δ_{ij}, to get the $\hat{d}_{ij}$ (disparities),[7]
4) compute a stress (goodness-of-fit) statistic based on (d_{ij}) and $(\hat{d}_{ij})$, such as Eq. (2),
5) move the configuration $\mathcal{C}$ in a suitable direction to reduce stress,

— loop through 1)−5) until no further improvement is possible.[8]

[7] Thus, "disparities are scaled to match the configuration distances (d_{ij}) as closely as possible". (Manly)
[8] In the iterative minimization process, care must be taken to avoid local minima, see Stata manual for details.

More about Multivariate Distances

Many different measures of point distance (dissimilarity) exist, and our focus here is on measures for other data types; first a couple of remarks:

- distances between variables $\sim$ transposing the data/formulation,
- similarity is "inverse" distance/dissimilarity,[9]
- many distance measures give equal weight to all variables $\Rightarrow$ some scaling may be necessary/meaningful if variables are not on same (similar) scales.

Binary data leads to 2×2-matrices/tables, say $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ of counts for agreement patterns between two points $(\mathbf{x}, \mathbf{y})$ across the p variables[10], e.g.

- matching proportion: $d(\mathbf{x}, \mathbf{y}) = (a+d)/(a+b+c+d) = (a+d)/p$,
- Jaccard distance: $d(\mathbf{x}, \mathbf{y}) = a/(a+b+c)$.

Mixed data involves balancing continuous and discrete distances, e.g. by averaging deviations for each variable j computed as (in the "Gower coefficient"),

- indicator (0/1) of agreement, for binary (possibly also categorical) variables,
- absolute deviation normalized by maximal deviation across entire dataset ($\sim$ scaled L_1-distance), for continuous variables.

[9] For example, for similarities $0 \leq s_{ij} \leq 1$ one may use $d_{ij} = 1 - s_{ij}$.

[10] For example, $a = \#$ variables (j) where $(x_j, y_j) = (1,1)$; $b = \#$ where $(x_j, y_j) = (1,0)$; $c = \#$ where $(x_j, y_j) = (0,1)$; $d = \#$ where $(x_j, y_j) = (0,0)$.

Hierarchical Clustering: Introduction

The idea:
to group observations ("points") that are close in distance together in clusters and display these in a way that reflect the distances.

First step:
combine two closest observations into a cluster (if multiple choices exist, pick one of these).

Next step:
redefine/recompute distance between points and cluster#, and combine two closest observations/clusters into new cluster.

Following steps:
redefine/recompute distances involving clusters (both to points and clusters)#, and combine two closest observations/clusters into new cluster.

Continue until only a single cluster is left.

We will need new distance definitions, namely *i)* between point and cluster, and *ii)* between two clusters — many possibilities exist (termed linkages; next page[11])

* not clear that any of these are better/more valid than others for all applications,
* focus here on single linkage, complete linkage and average linkage.

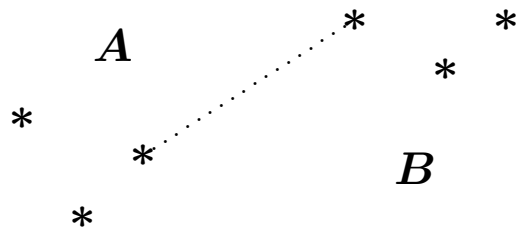
[11] See also Moodle site for detailed lists (from ED and Stata manual) of terminology and characteristics of linkages.

Selected Cluster Linkage Definitions

Definitions for clusters A and B (possibly consisting of a single point, but shown here with $\#A = 3$ and $\#B = 3$):

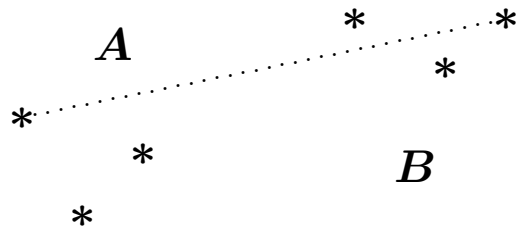
Single linkage:

$$d(A, B) = \min_{x \in A, y \in B} d(x, y)$$



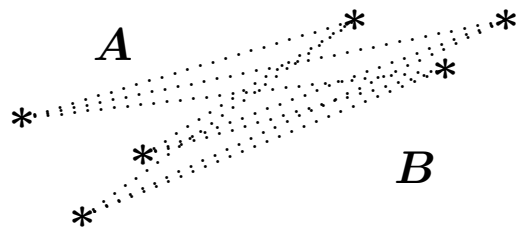
Complete linkage:

$$d(A, B) = \max_{x \in A, y \in B} d(x, y)$$



Average linkage:

$$d(A, B) = \frac{1}{\#A \cdot \#B} \sum_{x \in A, y \in B} d(x, y)$$



GM Farms Example: Single Linkage

(1): sea-way distances among $(3, 270, 408, 413, 491)$:

Farm	413	270	491	408	3
413	0	–	–	–	–
270	2.94	0	–	–	–
491	3.81	2.08	0	–	–
408	5.69	3.95	1.87	0	–
3	4.21	1.62	3.70	4.59	0

— smallest distance: farms $(3, 270)$: $d = 1.62$ (first cluster distance),

(2): recomputed distances (in blue):

Farm(s)	413	(3,270)	491	408
413	0	–	–	–
(3,270)	2.94	0	–	–
491	3.81	2.08	0	–
408	5.69	3.95	1.87	0

— smallest distance: farms $(408, 491)$: $d = 1.87$ (2^{nd} cluster distance),

(3): recomputed distances (in blue):

Farm(s)	413	(3,270)	(408,491)
413	0	–	–
(3,270)	2.94	0	–
(408,491)	3.81	2.08	0

— smallest distance: clusters $(3, 270)$ and $(408, 491)$: $d = 2.08$ (3^{rd} cluster distance),

(4) only farm 413 and cluster $(3, 270, 408, 491)$ left $\Rightarrow$ done!
and recomputed distance $d = 2.94$ (4^{th} cluster distance).

GM Farms Example: Average Linkage

(1): sea-way distances among $(3, 270, 408, 413, 491)$:

Farm	413	270	491	408	3
413	0	–	–	–	–
270	2.94	0	–	–	–
491	3.81	2.08	0	–	–
408	5.69	3.95	1.87	0	–
3	4.21	1.62	3.70	4.59	0

— smallest distance: farms $(3, 270)$: $d = 1.62$ (first cluster distance),

(2): recomputed distances (in blue):

Farm	413	(3,270)	491	408
413	0	–	–	–
(3,270)	3.575	0	–	–
491	3.81	2.89	0	–
408	5.69	4.27	1.87	0

e.g., $3.575 = (2.94 + 4.21)/2$.

— smallest distance: farms (408,491): $d = 1.87$ (2^{nd} cluster distance),

(3) recomputed distances (in blue):

Farm	413	(3,270)	(408,491)
413	0	–	–
(3,270)	3.575	0	–
(408,491)	4.75	3.58	0

— smallest d: farm 413 and cluster (3,270): $d = 3.575$ (3^{rd} cluster distance),

(4) only clusters (3,270,413) and (408,491) left $\Rightarrow$ done!
and recomputed distance $d = 3.97$ (4^{th} cluster distance).

The Dendrogram

- main tool for visualizing the clusters obtained from cluster analysis,
- multiple ways of drawing it (e.g. for horizontal order of objects),
- with many objects, it may be useful to focus on top parts only,
- exploratory in nature, but bootstrapping may be added to allow a sense of how stable the clusters are.[12]

Interpretation of the vertical axis: usually[13] the distance (or dissimilarity) between clusters (according to chosen distance measure and linkage method),

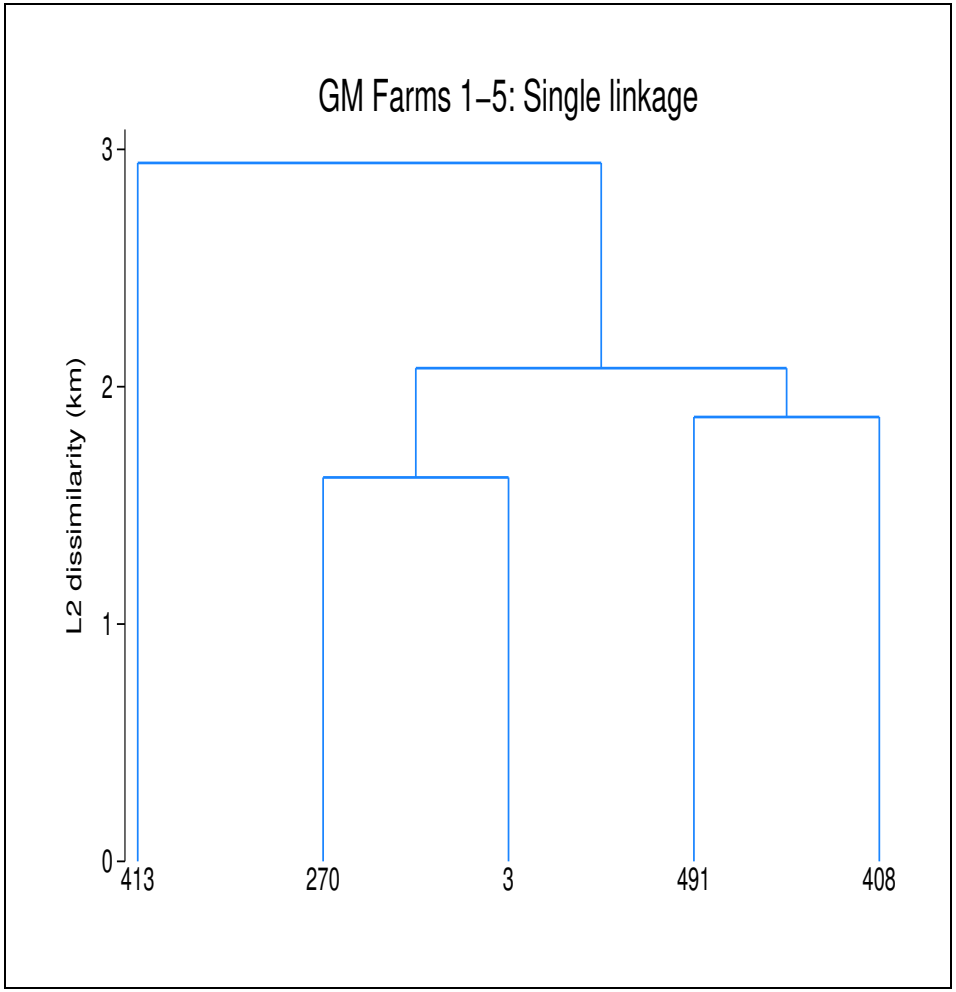
* horizontal lines $\sim$ clusters joined at that distance,
* no reconstruction of individual distances, only cluster distances,[14]
* large vertical distances $\sim$ relatively strong separation between clusters; short distances $\sim$ similar clusters (i.e., branches that could be joined).

Assessment of "fit" by hierarchical clustering: one idea[15] is to compare how individual (point) distances compare with cluster proximities using a simple correlation coefficient[16].

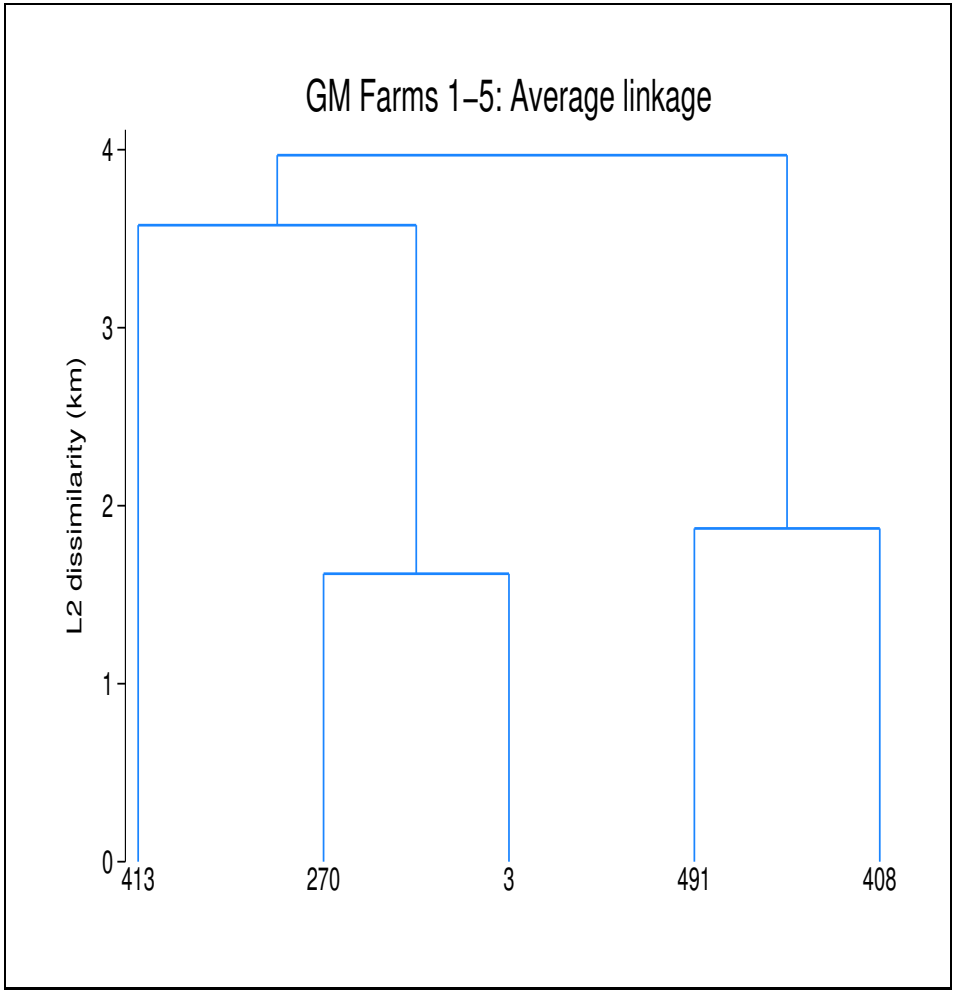
[12] See e.g. Hennig (2007), *Comput. Statist. Data Analysis* 52, 258−271; implemented in some R libraries.
[13] Also possible to display the similarity between clusters, defined suitably (in practice software-dependent).
[14] Some related diagrams (e.g., phylogram, phylogenetic tree) represent actual distances.
[15] For more discussion, see e.g. Arbelaitz et al. (2013), *Pattern Recognition* 46, 243−256, or Hennig (2015), *Pattern Recognition Letters* 64, 53−62.
[16] Also termed cophenetic correlation, for details see e.g. the textbook by Everitt et al., 2011, *Cluster Analysis*.

Dendrograms for GM Cluster Analyses

Single linkage dendrogram (Stata):

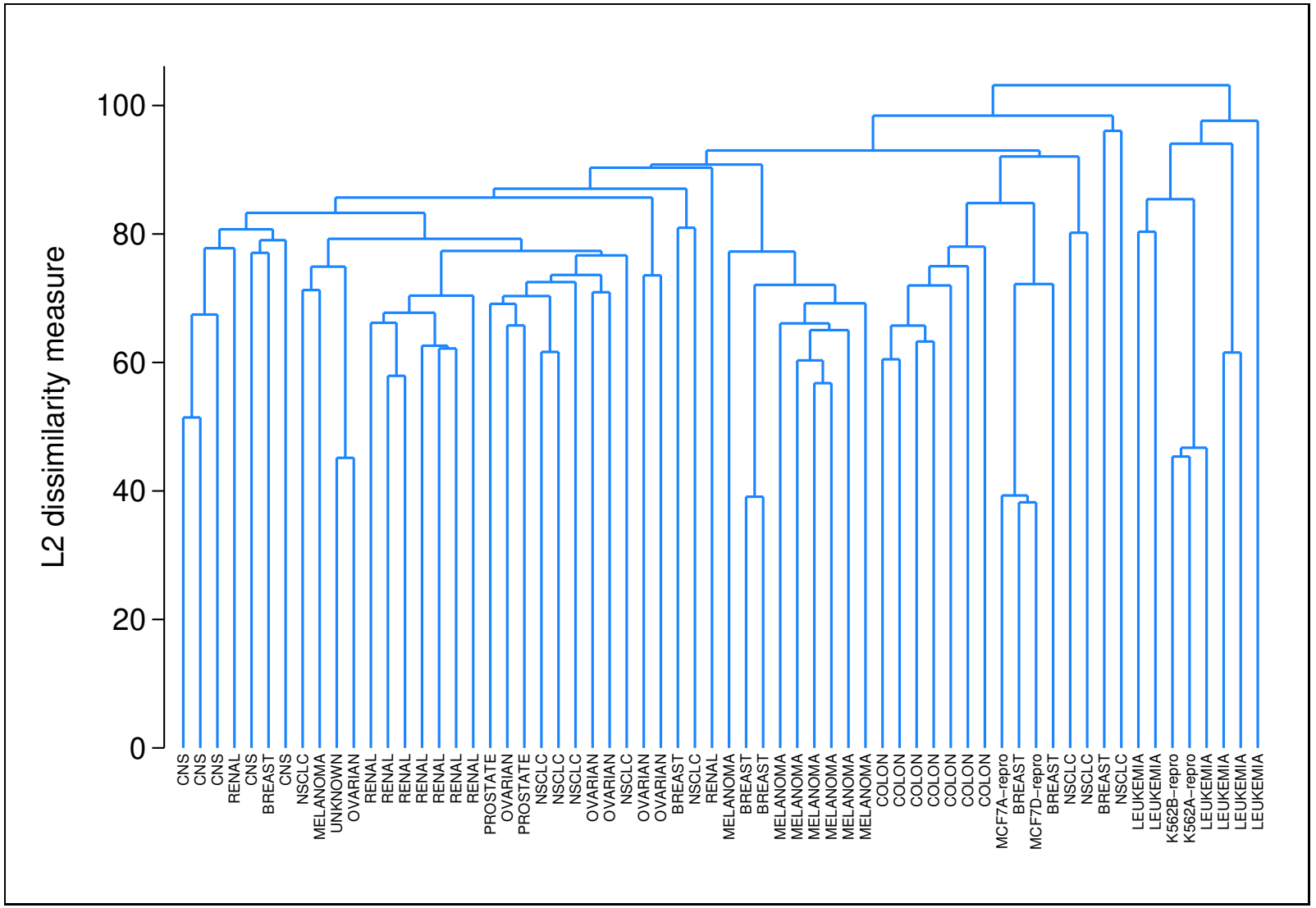


Average linkage dendrogram (Stata):



Hierarchical Clustering for NCIARRAY Data

Average linkage dendrogram (from Stata) $\sim$ SL Figure 10.17:



The different cancer types in the cell lines are reflected in the dendrogram.

K-Means Clustering

Perhaps best known example of partition-based clustering methods, mathematically defined as the solution to the problem of

finding a partition $C_1, \ldots, C_K$ of all (p-dimensional) points (x_i) so that

– every point x_i belongs to exactly one set C_k,

– the partition minimizes the sum of within-cluster sum of squares[17], i.e.,

$$\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2, \quad \text{where } \bar{x}_{kj} = \sum_{i \in C_k} x_{ij} / \#C_k \,.$$

Comments and interpretations:

- $\bar{x}_{kj}$ is the mean of the j^{th} component (e.g., variable) in the k^{th} cluster C_k,
- K is fixed for the problem/algorithm (a major difficulty is the choice of K),
- finding the global minimum of the criterion is a very large problem, but iterative algorithms starting from a suitable starting point can find local minima,
  - different starting points (and algorithms) may, or rather will, find different solutions, so strongly recommended (SL) to try out a range of searches,
  - assessment of the solution is necessary (no consensus on validation).

[17] The criterion can also be represented as the average squared Euclidean distances between points within each cluster, summed across clusters.

Iris Data: K-means

The known grouping into species can be used to explore the performance of K-means algorithms; obviously we take $K = 3$.

Stata implementation — 10 runs with random starting configurations, and largest species proportions in clusters:

Proportion	Run (raw data)			Run (standardized)					
Species	1 (×5)	2 (×3)	3 (×2)	1 (×4)	2	3 (×2)	4	5	6
Iris setosa	.64	1.0	1.0	1.0	.98	1.0	.98	1.0	.66
Iris versicolor	.92	.96	.94	.78	.74	.76	.74	.76	.92
Iris virginica	1.0	.72	.72	.66	.84	.72	.84	.78	1.0
% variation explained	79.0	88.4	88.4	76.7	76.5	76.7	76.5	76.7	68.2

— seems to perform better on standardized variables, but which is best??

R implementation — same setup as above:

Proportion	Run (raw data)			Run (standardized)		
Species	1 (×5)	2 (×5)	best 100	1 (×8)	2 (×2)	best 100
Iris setosa	.66	1.0	1.0	1.0	.66	1.0
Iris versicolor	.92	.96	.96	.78	.92	.78
Iris virginica	1.0	.72	.72	.72	1.0	.72
% variation explained	79.0	88.4	88.4	76.7	68.2	76.7

— seems to be less variable than in Stata and to perform best on raw data.

Interpretation: results support SL recommendation to explore multiple settings/runs!