Scientific Computing for Biologists

Lecture 8: (Dis)similarity and clustering

Instructor: Paul M. Magwene

01 November 2011

Outline of Lecture

- Distance and dissimilarity measures
  - Quantitative data
  - Dichotomous data
  - Qualitative data
- Hierarchical clustering
- Neighbor-joining
- Multidimensional scaling (MDS)
- Minimum Spanning Tree (MST)

Similarity/Dissimilarity

Intuition

Similarity is a measure of "likeness" between two entities of interest. Dissimilarity is the complement of similarity.

- Dissimilarities may be converted to similarities (and vise versa) by taking any monotonically decreasing function. For example:

$$s = 1 - d_{ij} \text{ (for } 0 \leq d_{ij} \leq 1)$$

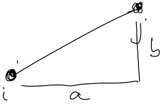
- Dissimilarities are usually in range $0 \leq d_{ij} \leq C$ where C is the maximum dissimilarity
- Distances are one measure of dissimilarity but distances are unbounded to the right

$$d_{ij} \in [0, \infty]$$

Dissimilarity Measures for Quantitative Data

- Euclidean Distance

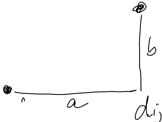
$$d_{ij} = \left\{ \sum_{k=1}^{p} (x_{ik} - x_{jk})^2 \right\}^{1/2}$$



$d_{ij} = \sqrt{a^2 + b^2}$

- Manhattan (taxi-cab) distance

$$d_{ij} = \sum_{k=1}^{p} | x_{ik} - x_{jk} |$$



$d_{ij} = a + b$

- Scaled Euclidean Distance

$$d_{ij} = \left\{ \sum_{k=1}^{p} w_k^2 (x_{ik} - X_{jk})^2 \right\}^{1/2}$$

where W_k are suitable weights

e.g. $\left(\text{std. dev of variable } k\right)^{-1}$ or $\left(\text{range of } k\text{th variable}\right)^{-1}$

Metric vs. Non-metric

A non-negative function, $g(x,y)$, is metric if:

i) Satisfies the triangle inequality:

$$g(x,y) \leq g(x,z) + g(y,z)$$

ii) Symmetric:

$$g(x,y) = g(y,x)$$

iii) $g(x,y) = 0$ only if $x = y$

Euclidean Dist. is a metric function
(as is Manhattan distance)

Other Quantitative Measures of Dissimilarity

- Minkowski Metric

$$d_{ij} = \left\{ \sum_{k=1}^{p} |x_{ik} - x_{jk}|^{\lambda} \right\}^{1/\lambda} \quad \text{for integers } \lambda$$

$\lambda=1$ is Manhatten distance, $\lambda=2$ is Euclidean Dist.

- Canberra Metric

$$d_{ij} = \sum_{k=1}^{p} \frac{|x_{ik} - x_{jk}|}{(x_{ik} + x_{jk})}$$

[Accts for distance btw. points & relationship to origin]

→ only for non-negative values

- Czekanowski Coefficient

$$d_{ij} = 1 - \frac{2 \sum_{k=1}^{p} \min(x_{ik}, x_{jk})}{\sum_{k=1}^{p} (x_{ik} + x_{jk})}$$

[% dissimilarity over all variables]

Quantitative Dissimilarity for Variables

Correlation provides a suitable measure of similarity

$d_{kl} = 1 - r_{kl}$ if $r_{kl} = -1$ is taken to indicate maximum disagreement

$d_{kl} = 1 - r_{kl}^2$ is appropriate if $r_{kl} = 1$ and $r_{kl} = -1$ are treated equivalently (predictive power)

$$d_{kl} = 1 - \frac{\sum_{i=1}^{n} x_{ik} x_{il}}{\left(\sum_{i=1}^{n} x_{ik}^2 \sum_{i=1}^{n} x_{il}^2 \right)} \quad \leftarrow \text{uncentered correlation}$$

Dissimilarity for Dichotomous Data

For each pair of objects of interest form a 2×2 contingency table

		Obj 2	
		+	−
Obj 1	+	a	b
	−	c	d

$a+b+c+d=p$

Simple Matching: $d_{ij} = 1 - \frac{a+d}{p} = \frac{b+c}{p}$

Jaccard Coefficient: $d_{ij} = \frac{b+c}{a+b+c}$ (joint absence does not contribute)

Czekanowski Coeff: $d_{ij} = \frac{b+c}{2a+b+c}$

Dissimilarity btwn. Variables

$a + b + c + d = n$ (# of objects/individuals)

a = # of objects showing + for both variables, k & l

... etc.

	+	−
+	a	b
−	c	d

$$\chi^2 = \frac{(ad-bc)^2(a+b+c+d)}{(a+b)(a+c)(c+d)(b+d)}$$

$$d_{kl} = 1 - \sqrt{\frac{\chi^2}{n}}$$

Dissimilarities for mixed data types

Gower (1971) suggests:

$$S_{ij} = \frac{\sum_{k=1}^{p} W_{ijk} S_{ijk}}{\sum_{k=1}^{p} W_{ijk}}$$

where S_{ijk} is the similarity for i & j based on variable k

- recommends

$S_{ijk} = 1$ for binary data w/positive match & categorical data when i and j in same category

$$S_{ijk} = 1 - \frac{|X_{ik} - X_{jk}|}{R_k}$$

for continuous variables where R_k is range of variable k

$W_{ijk} = 0$ when k missing on i or j

$W_{ijk} = W_k$ otherwise (often 1)

Define dissimilarity as:

$$d_{ij} = (1 - S_{ij})^{1/2}$$

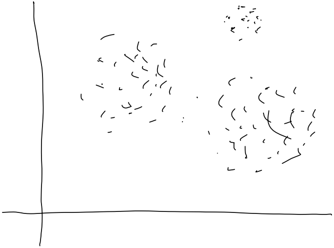
Introduction to Clustering

Goal of Clustering

- Find "natural groups" in data

→ one definition:

patches of high dimensity surrounded by patches of lower density in the p-dimensional space defined by the variates

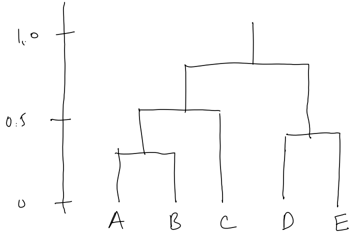


Hierarchical Clustering

Agglomerative / Divisive methods

- In practice almost always agglomerative

For n data points define a set of n-1 joins that represent groupings of objects @ different levels of similarity



Simple Algorithm for Hierarchical Clustering

1) Calculate a dissimilarity matrix for the n items

2) Join the two nearest items, i & j

3) Delete the ith & jth row and column of the dissimilarity matrix; add a new row/column

* that represents dissimilarity of new group (i;j) to all other items

4) Repeat from step 2 until there is a single group

Methods of Hierarchical Clustenug

The different methods are determined by the function used to determine the distance between groups

Some Common Group Distance Criteria

Single linkage (nearest neighbor)

Complete linkage (furthest neighbor)

Group average

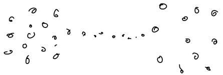
Centroid

Single Linkage Clustering

n_i, n_j are # of objects in groups i & j

☆ D_{ij} is the smallest of the $n_i n_j$ dissimilarties between each element of i & each element of j

- → Invariant under monotonic transformation of the d_{ij}
- → Unaffected by ties
- → Provably nice assymptotic properties
- → Susceptible to "chaining"



Complete Linkage

D_{ij} is the maximum of the $n_i n_j$ dissimilarities between the two groups

→ also invariant under monotonic transformation

Group average

D_{ij} is the average of the $n_i n_j$ dissimalarities between the two groups (UPGMA, WPGMA)

Centroid method

D_{ij} is the squared euclideen distance between the centroids of groups i & j

Hierarchical Clustering, A worked Example

	A	B	C	D	E
A	0				
B	4	0			
C	$\boxed{1}$	4	0		
D	4	2	4	0	
E	5	5	3	4	0

Single Linkage

	(A, C)	B	D	E
(A, C)				
B	4			
D	4	2	0	
E	3	5	4	0

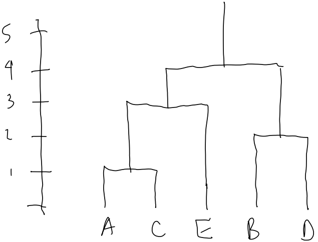
	(A, C)	(B, D)	E
(A, C)	0		
(B, D)	4	0	
E	$\boxed{3}$	4	0

Worked Example, cont.

	((A,C),E)	(B,D)
((A,C),E)	0	
(B,D)	4	0

→ Only one Choice

((A,C),E),(B,D))



Single Linkage Clustering

Neighbor Joining

Originally described by Saitou and Nei, 1987.

Goal

Tries to create the (unrooted) tree topology with the least branch length (minimum-evolution criterion).

Basic algorithm:

1. Calculate matrix Q (next slide) from the distance matrix
2. Find the pair of taxa in Q with the lowest value; create a node on the tree that joins these two taxa (i.e. the closest neighbors)
3. Calculate the distance of each of the taxa in the pair to this new node
4. Calculate the distance of all taxa outside of this pair to the new node
5. Repeat from step 1 using the distances calculated in the previous step

Neighbor Joining, cont.

$$Q_{ij} = (r-2)d_{ij} - (R_i + R_j)$$

where r is the number of taxa, d_{ij} is the distance between taxa i and j and R_k is the row sum over row k of the distance matrix ($R_k = \sum_i d_{ik}$).

When nodes i and j are joined they are replaced by a node, A, with distance to a remaining node k given by:

$$d_{Ak} = \frac{1}{2}(d_{ik} + d_{jk} - d_{ij})$$

NJ example from Saitou and Nei 1987

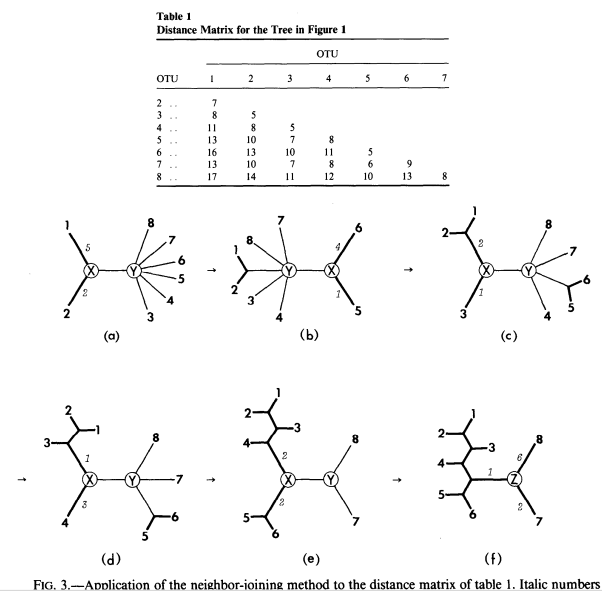


Table 1
Distance Matrix for the Tree in Figure 1

	OTU						
OTU	1	2	3	4	5	6	7
2 . .	7						
3 . .	8	5					
4 . .	11	8	5				
5 . .	13	10	7	8			
6 . .	16	13	10	11	5		
7 . .	13	10	7	8	6	9	
8 . .	17	14	11	12	10	13	8

FIG. 3.—Application of the neighbor-joining method to the distance matrix of table 1. Italic numbers

Multidimensional Scaling (MDS)

Goal

Given dissimilarities between objects, d_{ij}, estimate a k-dimensional set of points, $\mathbf{X}$, such that $|x_i - x_j| \approx d_{ij}$.

Derivation of MDS

Motivation

If we know the coordinates of n points in p-dimensional space, we can easily calculate the Euclidean distances between every pair of points. Can we reverse this process, starting with the distances and getting back the coordinates points?

Consider a data matrix $\mathbf{X}$ $(n \times p)$. Let $\mathbf{Q} = \mathbf{X}\mathbf{X}'$ be a $n \times n$ matrix, where

$$q_{rs} = \sum_{j=1}^{p} x_{rj} x_{sj}$$

If d_{rs}^2 is the squared Euclidean distance between points r and s then we can write this as:

$$\begin{aligned} d_{rs}^2 &= \sum_{j=1}^{p} (x_{rj} - x_{sj})^2 \\ &= q_{rr} + q_{ss} - 2q_{rs} \end{aligned}$$

Derivation of MDS, cont.

With a little bit of simple algebra we can show that:

$$q_{rs} = -\frac{1}{2}(d_{rs}^2 - d_{r.}^2 - d_{.s}^2 - d_{..}^2)$$

where a dot represent the average of values over the corresponding suffix: $d_{r.}^2$ is the average over the rth row of matrix $\mathbf{D} = (d_{ij}^2)$, $d_{.s}^2$ is the average over the sth column of $\mathbf{D}$, and $d_{..}^2$ is the average of all elements of $\mathbf{D}$.So, given $\mathbf{D}$, the squared interpoint distances, we can regenerate $\mathbf{Q}$.

Since $\mathbf{Q}$ is symmetric, we can use eigendecomposition to write $\mathbf{Q} = \mathbf{T\Lambda T'}$ where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues of $\mathbf{Q}$ and $\mathbf{T}$ is the matrix of eigenvectors. Furthermore we can write $\mathbf{Q} = \mathbf{T\Lambda T'} = \mathbf{T\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}T'} = \mathbf{XX'}$ where $\mathbf{X} = \mathbf{T\Lambda^{\frac{1}{2}}}$.

Thus we've found how to get $\mathbf{X}$ from the squared distances.

See Krzanowski, W. J. (2000) Principles of multivariate analysis, for full details.

Algorithm for MDS

Given an $n \times n$ matrix of dissimilarities, $\mathbf{D}$, with elements d_{ij}:

1. Form matrix, $\mathbf{E}$, where $e_{ij} = -\frac{1}{2} d_{ij}^2$
2. Subtract from each element of $\mathbf{E}$ the means of the row and column in which it is located and the mean of all elements of $\mathbf{E}$; call the resulting matrix $\mathbf{F}$
3. Calculate the eigenvalues (λ_i) and eigenvectors $\mathbf{v}_i$ of $\mathbf{F}$, sorted in decreasing order. Eigenvectors should be normalized (i.e. $\mathbf{v}_i \cdot \mathbf{v}_i = 1$).
4. The coordinates of the n point on the j-th axis are given $\sqrt{\lambda_j}\mathbf{v}_j$

Potential MDS Complications

If the d_{ij} are metric (i.e. $d_{ij} \leq d_{ik} + d_{kj}$) than **F** is always positive semidefinite (psd; i.e. eigenvalues ≥ 0).

If **F** is not psd than how do you handle negative eigenvalues?

- Most common approach is only to consider positive eigenvalues
- This is OK if negative eigenvalues have small magnitude
- If negative eigenvalues are large than approximation tends to be poor

Multidimensional Scaling: Keep in mind...

- The configuration produced by any MDS method is indeterminate with respect to translation, rotation, and reflection.

Relationship between metric MDS and PCA

If the d_{ij} are Euclidean distances from a data matrix, $\mathbf{X}$, then metric MDS of $\mathbf{D}$ yields the PC scores obtained by PCA of $\mathbf{X}$.

Interpretation

PCA and MDS are dual methods:

- One operates on variable space (PCA)
- The other operates on subject space (MDS)

Other Metric MDS Approaches

- Classical MDS minimizes:

  $$\sum_i \sum_j (\delta_{ij}^2 - d_{ij}^2)$$

  where δ_{ij} is the distance between observations i and j in the MDS approximation.
- Alternates approaches try to minimize other measures of discrepancy. For example, "Sammon MDS" minimizes:

  $$\sum_i \sum_j (\delta_{ij} - d_{ij})^2$$

Non-Metric MDS

Non-metric MDS approaches try to preserve only the rank order of the distances.

If

$$d_{i1,j1} < d_{i2,j2} < \cdots < d_{im,jm}$$

then

$$\delta_{i1,j1} < \delta_{i2,j2} < \cdots < \delta_{im,jm}$$

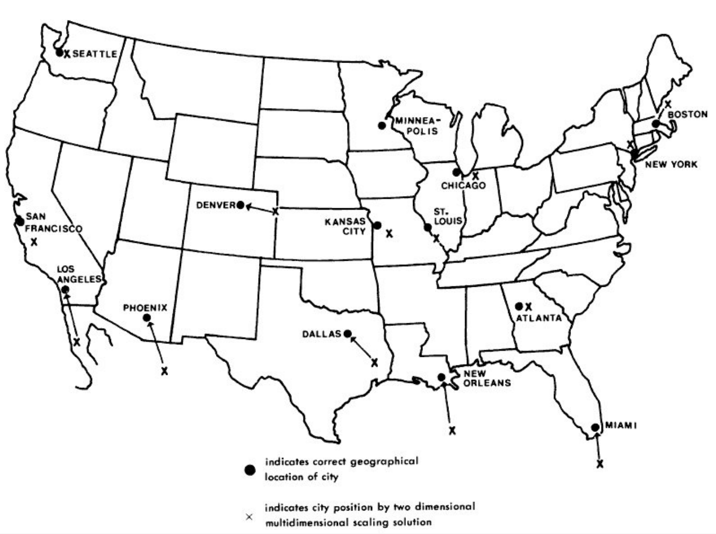
Shepard-Kruskal solution:

- Find $\hat{d}_{ij}$ that minimizes:

$$\text{STRESS} = \sqrt{\{\frac{\sum\sum_{i<j}(d_{ij} - \hat{d}_{ij})^2}{\sum\sum d_{ij}^2}\}}$$

MDS Example: Road Distances

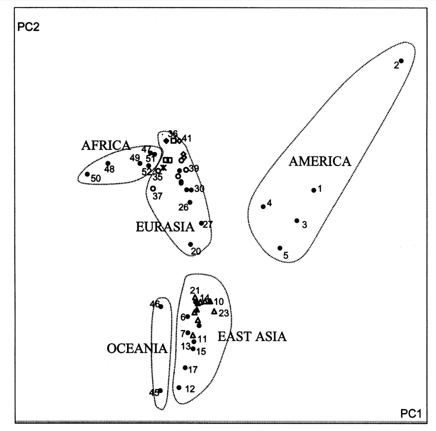
Input **D**: road distances between U.S. cities


SEATTLE
MINNEAPOLIS
BOSTON
NEW YORK
CHICAGO
DENVER
SAN FRANCISCO
KANSAS CITY
ST. LOUIS
LOS ANGELES
PHOENIX
ATLANTA
DALLAS
NEW ORLEANS
MIAMI
● indicates correct geographical location of city
× indicates city position by two dimensional multidimensional scaling solution

More MDS Examples I

Source: Zhivotovsky et al. (2003). Features of evolution and expansion of modern humans, inferred from genomwide microsatellite markers. Am J Hum Genet 72: 11711186.

Dissimilarities: F_{ST}'s between population samples.



Good MDS References

Kenkel, N. C. and L. Oroloci (1986). Applying metric and nonmetric multidimensional scaling to ecological studies: Some new results. Ecology 67:919-928

Minimum Spanning Tree

Goal

Construct a tree that connects all points in the data set and whose total length is minimized.

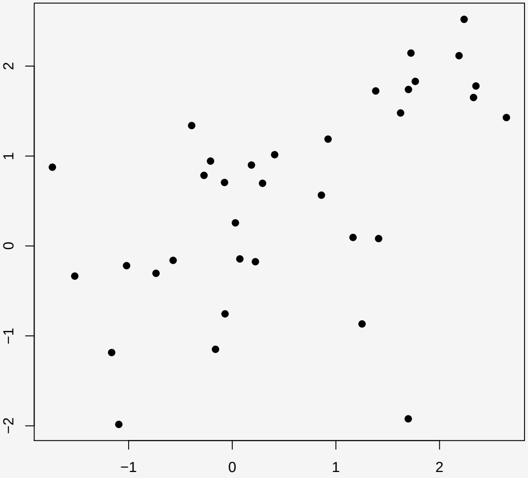
Statistical applications

- highlights close neighbors in a data set
- useful check for distortions produced by projection techniques
- tests of normality

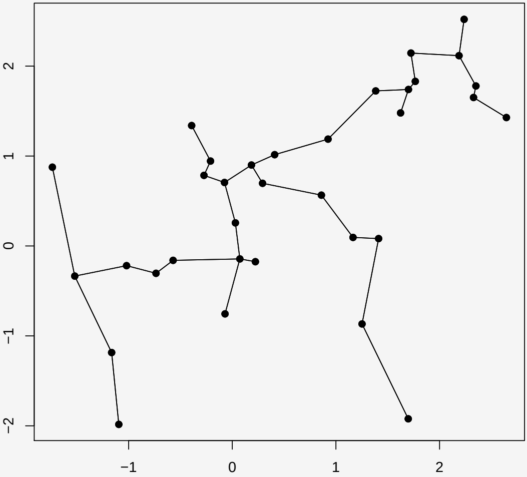
Other applications

- urban planning/engineering
- circuit design

Example Data Set



Minimum Spanning Tree: Example



Relationship between MST and Single Linkage Clustering

- Cut a single linkage dendrogram at height, $\delta \dashrightarrow$ clusters
- Remove all edges in the MST with length $\geq \delta \dashrightarrow$ subgraphs corresponding to the same clusters

A Generic MST Algorithm

Input: dissimilarity matrix, **D**, between each object (point) of interest

1. Create a graph, G, where $V = \{v_1, \ldots, v_n\}$ and $E = \{\}$ (E initially empty)
2. Find the smallest dissimilarity, d_{ij} where (i,j) is not in E.
3. Add (i,j) to E if (i,j) does not create a cycle
4. Repeat from step 2 until every vertex is included in at least one edge

Not particularly efficient algorithm, but simple. More efficient algorithms for finding MSTs include Kruskal's Algorithm and Prim's algorithm.

Applications of the MST

MST tends to highlight close neighbors; can be used to look for distortions associated with projections to lower dimensional spaces.

Using the MST to look for Projection Distortion

- Calculate the MST based on dissimilarity in a high-dimensional space
- Draw the MST edges among points in the projection space (e.g. MDS or PCA)
- MST edges that cross highlight geometric relationships among points that are not well represented by the projection