

# Scientific Computing for Biologists

## Lecture 10: Clustering I

Instructor: Paul M. Magwene

04 November 2014

# Outline of Lecture

- Distance and dissimilarity measures
  - Quantitative data
  - Dichotomous data
  - Qualitative data
- Hierarchical clustering
- Neighbor-joining
- 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 vice 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

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

- 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 weight for the  $k$ -th variable, e.g.  $\sigma_{x_k}^{-1}$  or  $(\max(x_k) - \min(x_k))^{-1}$

- Manhattan (taxi cab, city block) distance

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

## Dissimilarity Measures for Quantitative Data, cont.

- Manhattan (taxi cab, city block) distance

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

- Chebychev distance

$$d_{ij} = \max_k \{|x_{ik} - x_{jk}|\}$$

- Minkowski Metric

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

$\lambda = 1$  is Manhattan distance,  $\lambda = 2$  is Euclidean distance,  
 $\lambda = \infty$  is Chebychev distance.

# More distance measures

- Canberra distance (weighted Manhattan distance)

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

- Cosine distance

$$d_{ij} = \frac{x_{i\cdot} \cdot x_{j\cdot}}{|x_{i\cdot}| |x_{j\cdot}|}$$

where  $x_{i\cdot}$  and  $x_{j\cdot}$  indicate the row vectors, representing objects  $i$  and  $j$

- Hamming distance

$$d_{ij} = \frac{\text{count}(x_{ik} \neq x_{jk})}{p}$$

# Metric Distance Functions

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

- 1  $g(x, y)$  satisfies the triangle inequality:

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

- 2 symmetric:

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

- 3  $g(x, y) = 0$  only if  $x = y$

# Dissimilarity Measures for Dichotomous Data

For each pair of objects (samples) of interest form a  $2 \times 2$  contingency table:

	1	0
1	$a$	$b$
0	$c$	$d$

- Simple matching coefficient:

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

- Jaccard's coefficient (ignores joint absence):

$$d_{ij} = \frac{b + c}{a + b + c}$$

- Czenkanowski coefficient:

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



# Dissimilarity Measures for Variables

Correlation provides a suitable measure of *similarity*. Common *dissimilarity* measures based on correlation include:

- $d_{kl} = 1 - r_{kl}$  if  $r_{kl} = -1$  is taken to indicate maximum disagreement
- $d_{kl} = 1 - r_{kl}^2$  if  $r_{kl} = 1$  and  $r_{kl} = -1$  are treated equivalently (predictive power)
- Based on uncentered correlation:

$$d_{kl} = 1 - \frac{\sum_{i=1}^n x_{ik} x_{il}}{\sum_{i=1}^n x_{ik}^2 \sum_{i=1}^n x_{il}^2}$$

# Introduction to Clustering

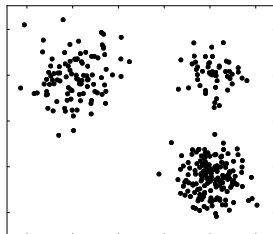
# Goal of Clustering

## Goal

Find “natural groups” in data

What’s a “natural group”?

- Patches of high density points surrounded by patches of lower density in the  $p$ -dimensional space defined by the variates.

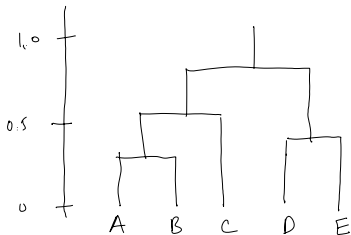


# Hierarchical Clustering

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



# Generic Algorithm for Agglomerative Hierarchical Clustering

- 1 Calculate a dissimilarity matrix for the  $n$  items
- 2 Join the two nearest items,  $i$  and  $j$
- 3 Delete the  $i$ -th and  $j$ -th rows and columns of the dissimilarity matrix; and a new row/column that represents the dissimilarity of a new group  $(i,j)$  to all other items
- 4 Repeat from step 2 until there is a single group

## Key Point

The different hierarchical clustering methods are determined by the function used to calculate the distance between groups in step 3.

# Single Linkage Clustering

## Group Distance Measure

Let  $i$  and  $j$  be groups, and  $n_i$  and  $n_j$  be the number of objects in the respective groups.

$D_{ij}$  is the *smallest* of the  $n_i n_j$  dissimilarities between each element of  $i$  and each element of  $j$

## Properties of Single Linkage Clustering

- Invariant under monotonic transformation of the  $d_{ij}$
- Unaffected by ties
- Provably nice asymptotic properties
- Disadvantage: susceptible to chaining

# Hierarchical Clustering, A worked Example

	A	B	C	D	E
A	0				
B	4	0			
C	①	4	0		
D	4	2	4	0	
E	5	5	3	4	0

Single Linkage

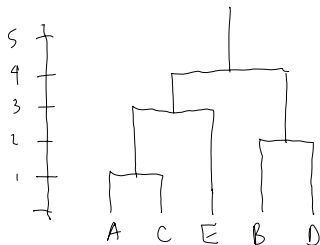
	(A,C)	B	D	E
(A,C)				
B	4			
D	4	2	0	
E	3	5	4	0

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



# More Hierarchical Clustering Functions

**Complete Linkage** –  $D_{ij}$  is the maximum of the  $n_i n_j$  dissimilarities between the two groups.

**Group Average Methods** –  $D_{ij}$  is the average of the  $n_i n_j$  dissimilarities between the two group (UPGMA, WPGMA)

**Centroid Method** –  $D_{ij}$  is the squared Euclidean distance between the centroids of groups  $i$  and  $j$

# 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

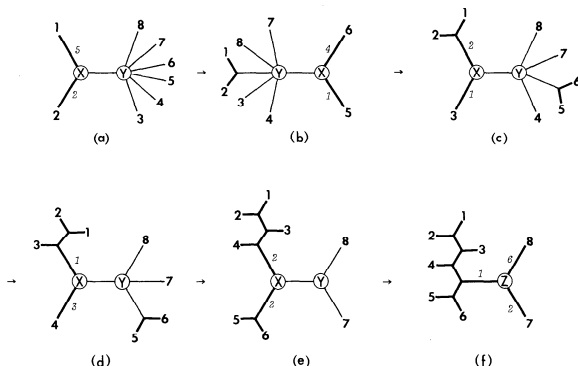


FIG. 3.—Application of the neighbor-joining method to the distance matrix of table 1. Italic numbers are branch lengths, and branches with thicker lines indicate that their lengths have been determined.

# Minimum Spanning Tree

## Goal

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

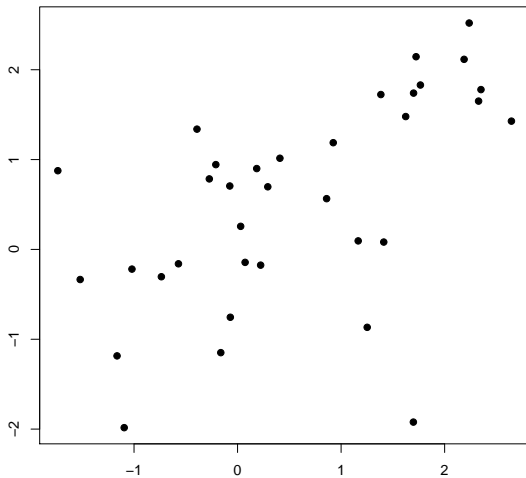
### *Statistical applications*

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

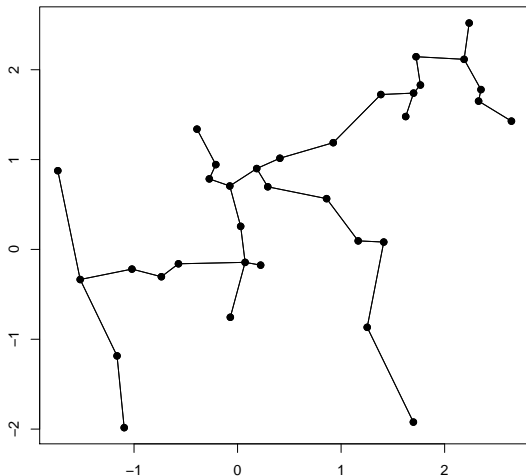
### *Other applications*

- 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,  $\delta \rightarrow$  clusters
- Remove all edges in the MST with length  $\geq \delta \rightarrow$  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  $V = \{v_1, \dots, v_n\}$  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