Basics of Data Clustering.

A. Govan

Department of Mathematics
North Carolina State University

MA 591R, March 2007

Outline.

Set up

- n patterns (objects to cluster)
- d features (description of an object)
  - type (binary, discrete, continuous)
  - scale (nominal, ordinal, interval and ratio)

Proximity index.

- pattern matrix, patterns by features, $n \times d$
- pattern matrix ($n \times d$) $\rightarrow$ proximity index $\rightarrow$ proximity matrix

Proximity Index

- similarity: larger - more in common (e.g. correlation coefficient)
- dissimilarity: larger - less in common (e.g. Euclidean distance)

More on proximity index

$d(i,k)$ - a proximity index between ith and kth patterns/objects and

1. $d(i,k) \geq 0$ for all i
2.
   (a) For dissimilarity: $d(i,i) = 0$, for all i.
   (b) For similarity: $d(i,i) \geq \max_k d(i,k)$, for all i.
3. $d(i,k) = d(k,i)$ for all (i,k)

Metrics (symmetric dissimilarity functions satisfying triangle inequality) also have

4. $d(i,k) = 0$ only if $i = k$
5. $d(i,k) \leq d(i,m) + d(m,k)$, for all (i,k,m)

Proximity index example

The Minkowski metric/distance (p-norm distance)
The $i$the pattern (ith row in the pattern matrix) be denoted as

$$\mathbf{x}_i = (x_{i1}x_{i2}...x_{id})^T$$

then the Minkowski metric is

$$d(i,k) = \left(\sum_{j=1}^{d} |x_{ij} - x_{kj}|^r \right)^{1/r} \qquad \text{where } r \geq 1$$

Special cases:

(a) Manhattan Distance, 1-norm

(b) Euclidean Distance, 2-norm

(c) Sup Distance, inf-norm ($r \to \infty$)

Data visualization.

Eigenvector Projection (Linear projection):

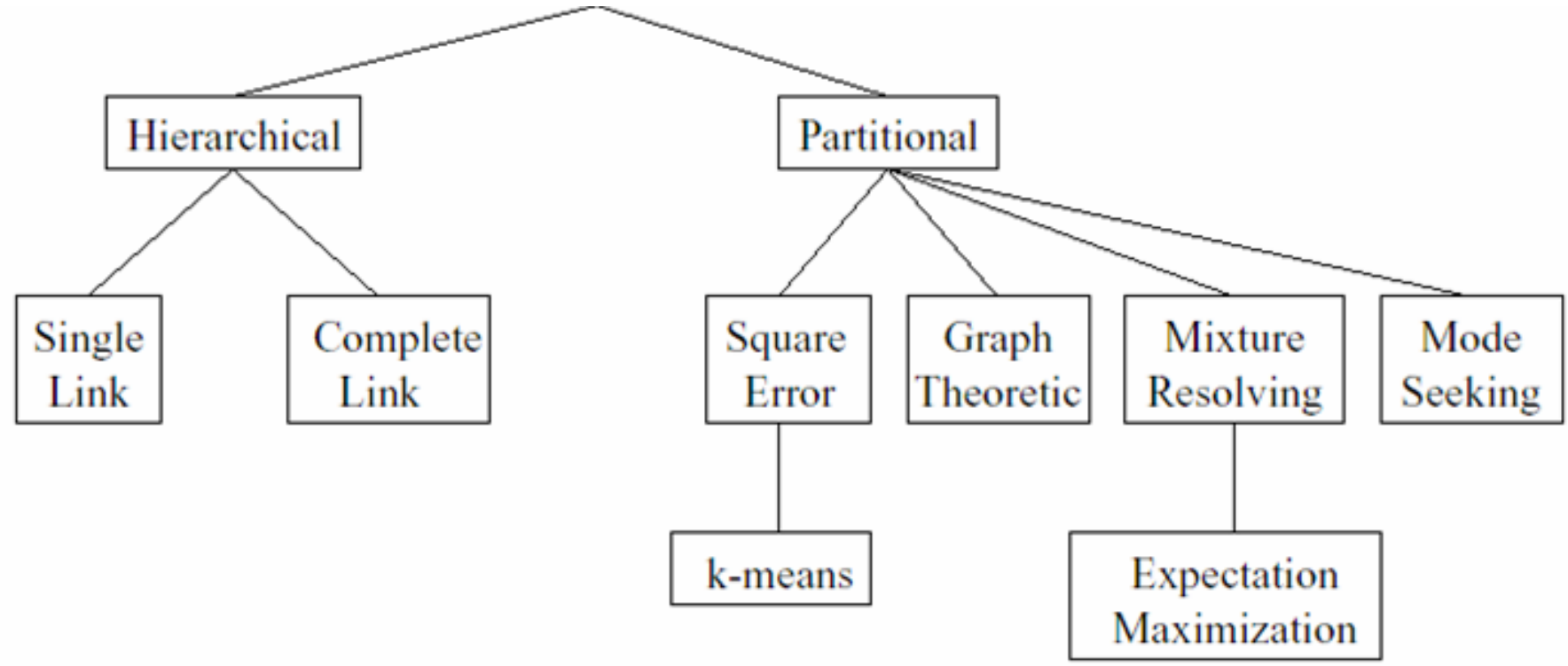
1. $\mathbf{M}$ is normalized pattern matrix (continuous features on a ratio scale) $m_{ij} = (x_{ij} - \mu_j)/s_j$
2. $\mathbf{R} = (1/n)M^T M$
3. $\mathbf{A^T} = \left[\begin{array}{ccc} \mathbf{c}_1 & \dots & \mathbf{c}_m \end{array} \right]$, where $\{\mathbf{c}_1, ..., \mathbf{c}_d\}$ are normalized eigenvectors of $\mathbf{R}$.
4. $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i$

Classification of Data.

- Non-Exclusive (overlapping)
- Exclusive
  - Extrinsic (supervised)
  - Intrinsic (Unsupervised)
    - Hierarchical - transforms a proximity matrix into a sequence of nested partitions.
    - Partitional (K-means) - single partition

Exclusive, intrinsic, partitional classification - **clustering**.
Exclusive, intrinsic, hierarchical classification - **hierarchical clustering**.

Intrinsic classifications.



Types of algorithms

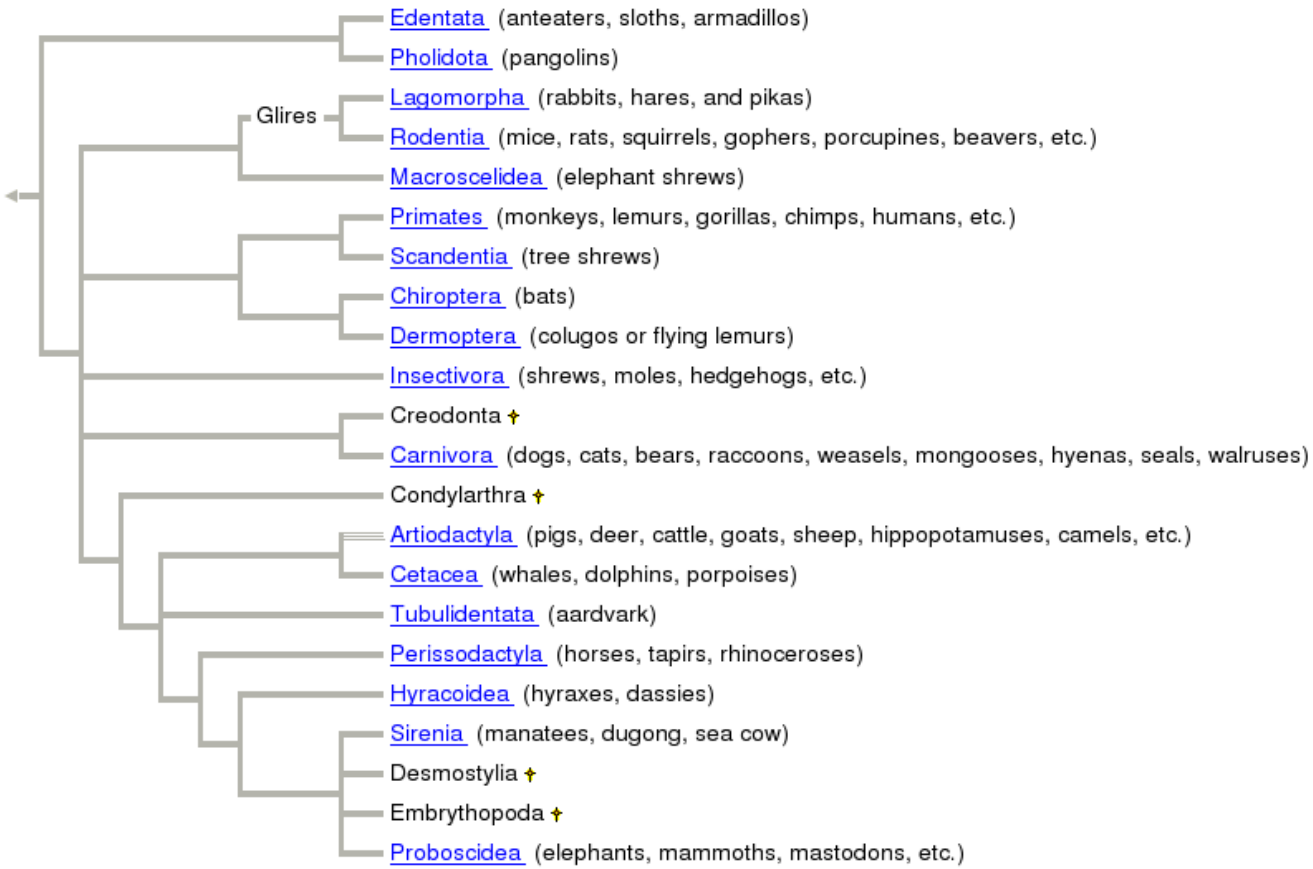
- Agglometrive, hierarchical classification; Divisive, hierarchical classification
- Serial; Simultaneous.
- Monothetic; Polythetic.
- Graph Theory; Matrix Algebra.

Dendrogram

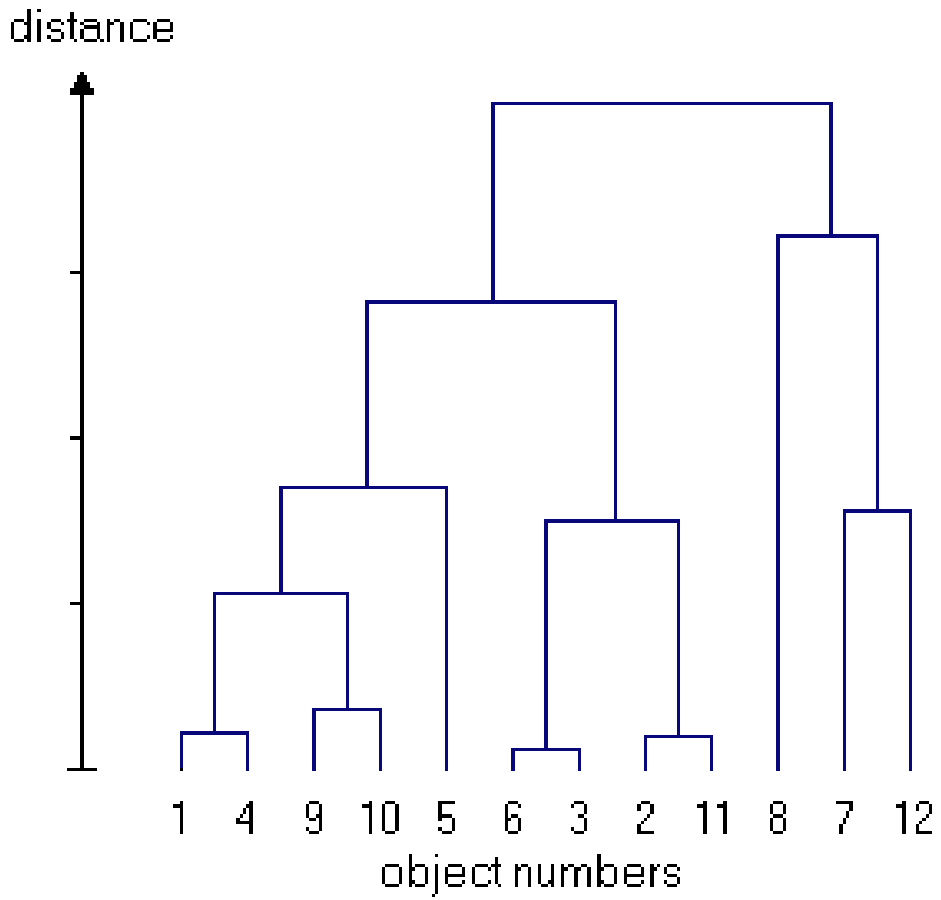
Dendrogram - special type of tree structure that provides a convenient picture of a hierarchical clustering.

- Threshold dendrogram
- Proximity dendrograms

Mammal evolutionary dendrogram.



Proximity dendrogram.



Hierarchical clustering methods.

- Single-Link Methods.
- Complete-Link Methods.

Single-Link Agglomerative algorithm.

Step 1. Begin with the disjoint clustering implied by threshold graph $G(0)$ which contains no edges and which places every object in a unique cluster, as the current clustering. Set $k \leftarrow 1$.

Step 2. Form threshold graph $G(k)$.
If the number of components (maximally connected subgraphs) in $G(k)$ is less then the number of clusters in the current clustering, redefine the current clustering by naming each component of $G(k)$ as a cluster.

Step 3. If $G(k)$ consists of a single connected graph, stop. Else, set $k \leftarrow k + 1$ and go to step 2.

Single-Link Johnson's Agglomerative algorithm.

Step 1. Begin with the disjoint clustering having level $L(0)$ and $m = 0$.

Step 2. Find the least dissimilar pair of clusters in the current clustering, $\{(r), (s)\}$, $d[(r), (s)] = min\{d[(i), (j)]\}$.

Step 3. $m \leftarrow m + 1$. Merge clusters (r) and (s), form the next clustering m. $L(m) = d[(r), (s)]$

Step 4. Deleting the rows and columns corresponding to clusters (r) and (s) and add a row and column corresponding to the newly formed cluster. The proximity between the new cluster (r, s) and the old cluster (k) is $d[(k), (r, s)] = min\{d[(k), (r)], d[(k), (s)]\}$

Step 5. If all objects are in one cluster, stop. Else, go to step 2.

Complete-Link Agglomerative algorithm.

Step 1. Begin with the disjoint clustering implied by threshold graph $G(0)$ which contains no edges and which places every object in a unique cluster, as the current clustering. Set $k \leftarrow 1$.

Step 2. Form threshold graph $G(k)$.
If two of the current clusters for a clique (maximally complete subgraph) in $G(k)$, redefine the current clustering by merging these two clusters into a single cluster.

Step 3. If $k = n(n-1)/2$, so that $G(k)$ is the complete graph on the n nodes, stop. Else, set $k \leftarrow k + 1$ and go to step 2.

Complete-Link Johnson's Agglomerative algorithm.

Step 1. Begin with the disjoint clustering having level $L(0)$ and $m = 0$.

Step 2. Find the least dissimilar pair of clusters in the current clustering, $\{(r), (s)\}$, according to $d[(r), (s)] = min\{d[(i), (j)]\}$.

Step 3. $m \leftarrow m + 1$. Merge clusters (r) and (s), form the next clustering m. $L(m) = d[(r), (s)]$

Step 4. Deleting the rows and columns corresponding to clusters (r) and (s) and add a row and column corresponding to the newly formed cluster. The proximity between the new cluster (r, s) and the old cluster (k) is $d[(k), (r, s)] = max\{d[(k), (r)], d[(k), (s)]\}$

Step 5. If all objects are in one cluster, stop. Else, go to step 2.

Evaluation of hierarchical clustering.

Strengths

- No need to specify number of clusters as input.
- can never undo what was done previously (nested clustering).

Weaknesses:

- do not scale well: time complexity of $O(n^2)$ or $O(n^2 \log n)$ (single-link, complete-link), where n is the number of total objects.
- can never undo what was done previously (nested clustering).

Square-Error Clustering Criteria.

- n patterns (objects) in d dimensions (features)
- K clusters $\{C_1, C_2, ..., C_K\}$
- Center of C_i: $\mathbf{m}^{(i)} = (1/n_i) \sum_{j=1}^{n_i} \mathbf{x}_j^{(i)}$
- Square-error of C_i: $\varepsilon_i^2 = \sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})^T (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})$
- Square-error for the entire clustering is $E_K^2 = \sum_{i=1}^{K} \varepsilon_i^2$

Algorithms for Clustering Data, Anil K. Jain and Richard C.Dubes, 1988.