COEN281 -- Introduction to Pattern Recognition and Data Mining

Lecture 15: Clustering (Part 2)

Instructor: Dr. Giovanni Seni
GSeni@scu.edu

Department of Computer Engineering
Santa Clara University

Fall/18

Syllabus

Week 1	Introduction; R (Ch.1)
Week 2	Bayesian Decision Theory (Ch.2; DHS: 2.1-2.6, 2.9) Parameter Estimation (DHS: 3.1-3.4)
Week 3	Linear Discriminant Functions (Ch.3&4; DHS: 3.8.2, 5.1-5.8) Regularization (Ch.6; SE: Ch.3)
Week 4	Neural Networks (DHS: 6.1-6.6, 6.8);
Week 5	Support Vector Machines (Ch.9)
Week 6	Decision Trees (Ch. 8.1; DHS: 8.3; Ch 2 SE)
Week 7	Ensemble Methods (Ch. 8.2; SE: Ch 4, 5)
Week 8	Clustering (Ch. 10; DHS: 10.6, 10.7) **Clustering** (DHS: 10.9); How many clusters are there? (DHS: 10.10)
Week 9	Non-metric: Association Rules Collaborative Filtering
Week 10	Text Retrieval; Other topics

Overview

- Hierarchical Clustering
- Between-Cluster Distance
- Agglomerative Clustering
  - Dendogram
  - Ties
  - Algorithm
  - Reversals
- Divisive Clustering
  - Connection to graph methods
- Simultaneous Clustering of Objects and Variables

Hierarchical Clustering

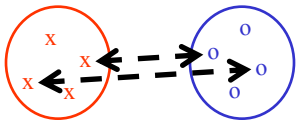
Overview

- No explicit notion of global score function
- Based on measures of distance between clusters
- Permit a convenient graphical display of clustering process: *dendogram*
- Agglomerative
  - Start with singletons, and successively merge clusters to form a nested sequence of partitions
- Divisive
  - Start with a single cluster of all points, and seek to split this into components

Distance Notion

Between Clusters

- Single Linkage: $d_{SL}(C_1, C_2) = \min_{\substack{x^i \in C_1 \\ x^j \in C_2}} \{d_{ij} = d(x^i, x^j)\}$
- Complete Linkage: $d_{CL}(C_1, C_2) = \max_{\substack{x^i \in C_1 \\ x^j \in C_2}} \{d_{ij} = d(x^i, x^j)\}$

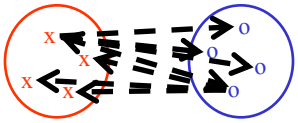


  - d_{SL} tends to produce long stringy clusters
    - Sensitive to perturbations and outlying points
  - d_{CL} tends to produce a few very compact (i.e., spherical) clusters
    - Appropriate for segmentation

Distance Notion

Between Clusters (2)

- Intermediate measure between *SL* and *CL* extremes…
  - Average Linkage: $d_{AL}(C_1, C_2) = \frac{1}{|C_1|}\frac{1}{|C_2|}\sum_{x^i \in C_1}\sum_{x^j \in C_2} d_{ij}$



- If you have the data vectors:
  - Centroid: $d(C_1, C_2) = dist(\hat{\mu}_{C_1}, \hat{\mu}_{C_2})$
  - Ward: $d^2(C_1, C_2) = \frac{2|C_1||C_2|}{|C_1|+|C_2|}\left\|\hat{\mu}_{C_1} - \hat{\mu}_{C_2}\right\|^2$

Distance Notion

Lance-Williams Formula

- Consider clusters C_i and C_j with $|C_i|=n_i$ and $|C_j|=n_j$
- If C_i and C_j merge -- i.e., $C_k = C_i \cup C_j$, is the distance $d(C_k,C_h)$ simply related to $d(C_h,C_i)$ and $d(C_h,C_j)$?

$$d_{hk} = \alpha_i \cdot d_{hi} + \alpha_j \cdot d_{hj} + \beta \cdot d_{ij} + \gamma \cdot \left| d_{hi} - d_{hj} \right|$$

	α_i	α_j	β	γ
d_{SL}	1/2	1/2	0	-1/2
d_{CL}	1/2	1/2	0	1/2
d_{Avg}	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
$(d_{Mean})^2$	$\frac{n_i}{n_i+n_j}$	$\frac{n_i}{n_i+n_j}$	$-\alpha_i\alpha_j$	0

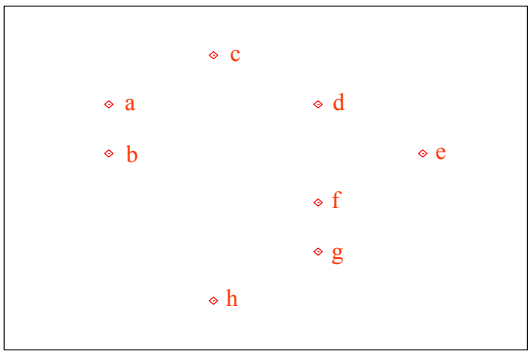
⇒ Provides a framework to parameterize clustering algorithms

Agglomerative Clustering

Example

- Consider

a	(0, 4)	c	(1, 5)	e	(3, 3)	g	(2, 1)
b	(0, 3)	d	(2, 4)	f	(2, 2)	h	(1, 0)



Agglomerative Clustering
Example (2)

- Complete linkage with squared Euclidean distance

d	b	c	d	e	f	g	h
a	1	2	4	10	8	13	17
b		5	5	9	5	8	10
c			2	8	10	17	25
d				2	4	9	17
e					2	5	13
f						1	5
g							2

⇨

	c	d	e	fg	h
ab	5	5	10	13	17
c		2	8	17	25
d			2	9	17
e				5	13
fg					5

⇩

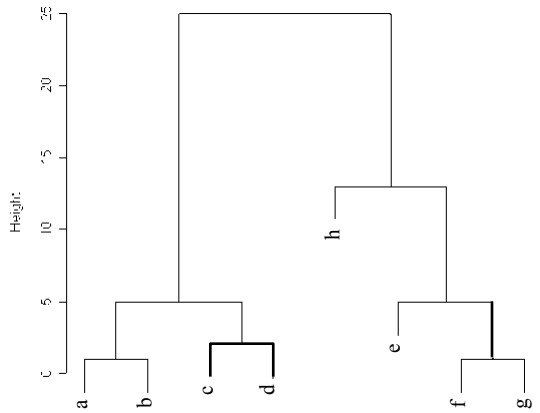
	cd	e	fg	h
ab	5	10	13	17
cd		8	17	25
e			5	13
fg				5

⇦

	efg	h
abcd	17	25
efg		13

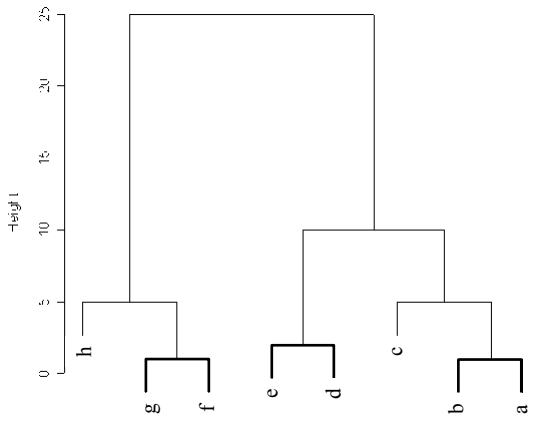
Agglomerative Clustering
Dendograms

- Shows sequence of merging
  - Height proportional to actual distances… large gap suggests clusters



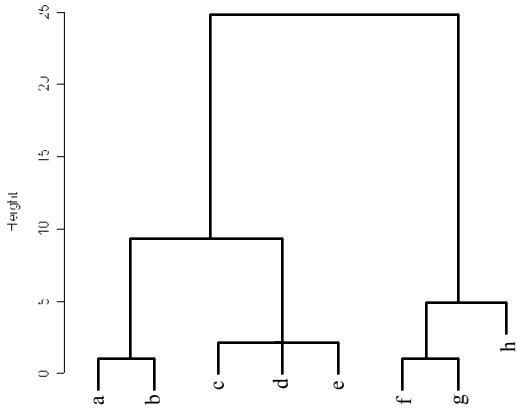
Agglomerative Clustering
The Issue of *Ties*

- Cluster output might depend on the order data was given!



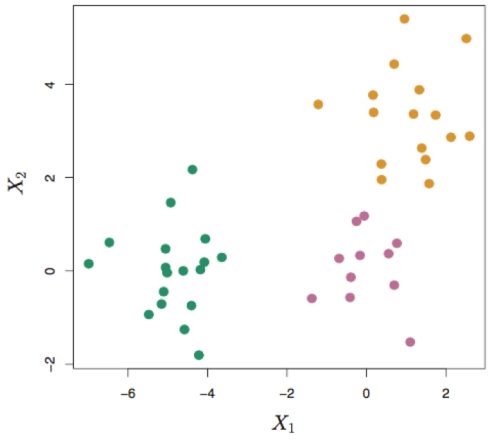
Agglomerative Clustering
The Issue of *Ties* (2)

- Jardine-Sibson technique
  - Make all possible merges at each level



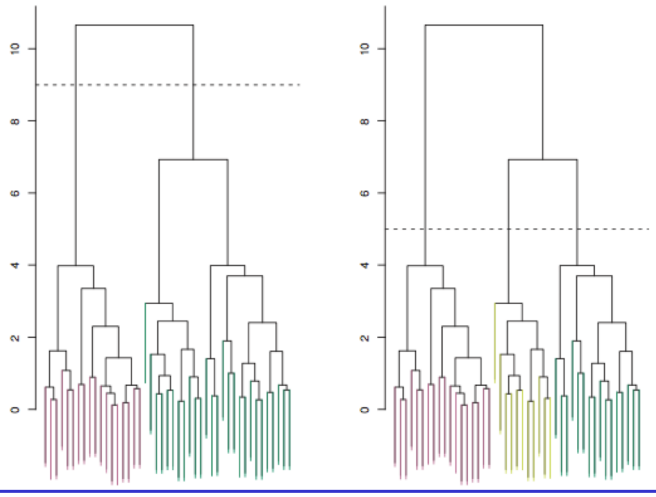
Agglomerative Clustering
Example

- 2D data... randomly generated with K=3



Agglomerative Clustering
Example (2)

- Horizontal cut across dendogram ==> K



Agglomerative Clustering
Basic Algorithm

Initialize n, $C_i = \{\boldsymbol{x}^i\}$

do

  let C_i and C_j be the clusters

    minimizing the distance between all pairs of clusters

  $C_i = C_i \cup C_j$

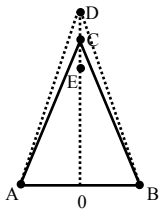
  removed cluster C_j

until *there is only one cluster left*

- $O(n^3)$ time and $O(n^2)$ space
  - All pairwise distances are needed at the start (but not the points themselves)
  - Hybrid algorithm?

Agglomerative Clustering
The Issue of *Reversals*

- It sometimes happens that when clusters A, B are merged, the distance $d(A \cup B, C) < \min\{d(A,C),\ d(B,C)\}$
  - This would make a dendogram hard to interpret – we expect increasing height to mean increasing distance



$A(-1/2, 0)$, $B(1/2, 0)$ and $C(0, Q)$ with $7/8 < Q < 1$

Using SE, $d(A,B) = 1$, and $d(A,C) = d(B,C) = Q^2 + 1/4 > 1$

Centroid of $A \cup B$ is at origin. Then, $d(A \cup B, C) = Q^2 < 1$

  - d_{Avg}, d_{SL}, d_{CL}, d_{Ward} don't result in reversals

Divisive Clustering
Overview

- Top-down, splitting
  - Start with one cluster of n objects
    Split it into two
    Recursively split
- Computation needed from one level to another is more intensive
- For an n element set, there are $2^{n-1}-1$ partitions with two elements!
  - Need heuristic to find "good" partitions
- *Monothetic* – split using one variable at a time
  - Limits the number of possible partitions that need to be examined

Divisive Clustering
Overview (2)

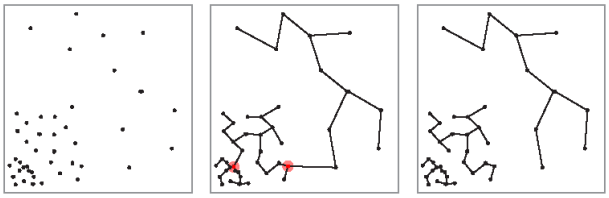
- *Polythetic* – split on the basis of all the variables
  - Which cluster to split? One with the largest *diameter*
    $diameter(C) = \max_{x^i, x^j \in C} d_{ij}$
  - To divide selected C
    - Find most disparate observation – i.e., x^i with $\max_{x^j \in C} d_{ij}$
    - $\boldsymbol{x}^i$ initiates the "splinter group"
    - Reassign observations that are closer to the "splinter group" than to the "old party"

      i.e., any j with $d_{ij} < \underset{x^l \neq x^i, x^j}{avg} d_{jl}$

Divisive Clustering

Connection to Graph-based Methods

- Similarity matrix and graph
  - matrix: $\mathbf{S}_{n\times n} = [s_{ij}]$, with $s_{ij} = \begin{cases} 1 & \text{if } s(x^i, x^j) > s_0 \\ 0 & \text{otherwise} \end{cases}$
  - graph: $G(S)=(V,E)$ -- nodes correspond to $\boldsymbol{x}^i$'s; $(i,j) \in E$ if $S_{ij}=1$
  - A Minimum Spanning Tree is $T \subseteq E$ with $\min \sum_{(u,v)\in T} d_{uv}$
- Removal of "inconsistent" edges
  - Ones with length significantly larger than the average incident upon a node



Simultaneous Clustering

Of Rows and Columns

- Analysis of voting data

	Republican vote (%) per Year					
State	1932	1936	1940	1960	1964	1968
Kentucky (KY)	40	40	42	54	36	44
Louisiana (LA)	7	11	14	29	57	23
Maryland (MD)	36	37	41	46	35	42
Mississippi (MI)	4	3	4	25	87	14
Missouri (MO)	35	38	48	50	36	45
South Carolina (SC)	2	1	4	49	59	39

  - One is interested not only in the structure within the set of objects (states) but also the structure within the set of variables (years)

Simultaneous Clustering
Of Rows and Columns (2)

- Dendograms for state and years
  - Additional processing to partition data matrix into "similar" blocks