Primary Objective behind Clustering

• Cluster Analysis ("data segmentation") is an exploratory method for identifying homogenous groups ("clusters") of records
• Similar records should belong to the same cluster
• Dissimilar records should belong to different clusters

Example: Fitting the Troops (from Data Mining Techniques by Berry & Linoff)

• The US army recently commissioned a study on how to redesign the uniforms of female soldiers. The army's goal is to reduce the number of different uniform sizes that have to be kept in inventory while still providing each soldier with well-fitting khakis.
• Researchers Ashdown and Paal @ Cornell University designed a new set of sizes based on the actual shapes of women in the army. Unlike traditional clothing size systems, the new sizes are not an ordered set of graduated sizes where all dimensions increase together.
• Instead, they came up with sizes that fit particular body types (e.g., short-legged, small waisted, large-busted women with long torsos, average arms, broad shoulders, and skinny necks).

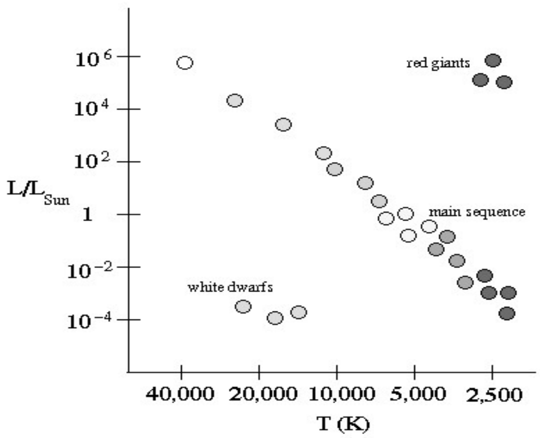
Investment:

Cluster securities based on financial performance info (return, volatility, beta) and other info (industry and market capitalization) to create a balanced portfolio

Industry Analysis:

For a given industry, cluster firms based on growth rate, profitability, market size, product range, presence in various international markets, to understand industry structure (determine competitors)

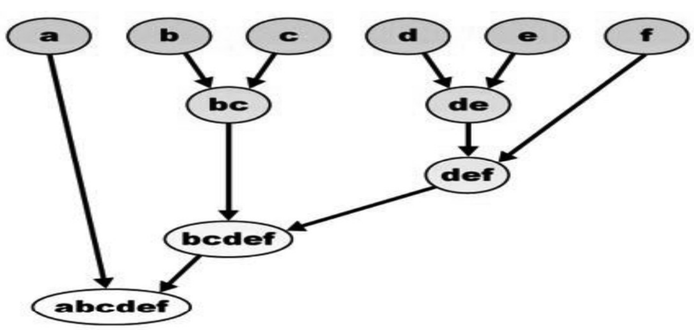
• Simple Clustering: 1-2 variables
• Visual inspection of data



Clustering > 2 Variables

• **Two approaches**:
– Compute "multivariate distance" between records, and group "close" records.
– Group records to increase within-group homogeneity.

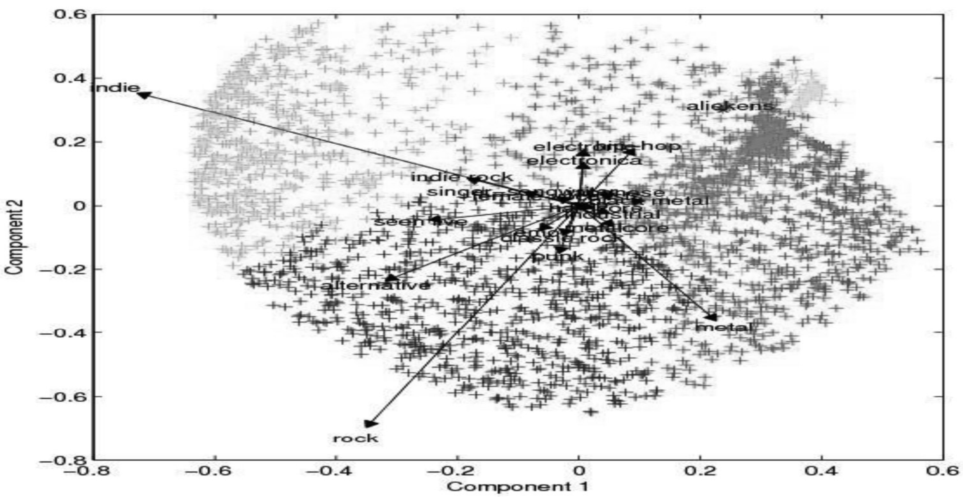
Two Types of Clustering Algorithms



Hierarchical methods - **agglomerative**:

Begin with n clusters; sequentially merge similar clusters until cluster is left.

- Useful when goal is to arrange the clusters into a natural hierarchy.
- Requires specifying distance measure.



Non-Hierarchical methods:

Pre-specified number of clusters assign records to each of the clusters.

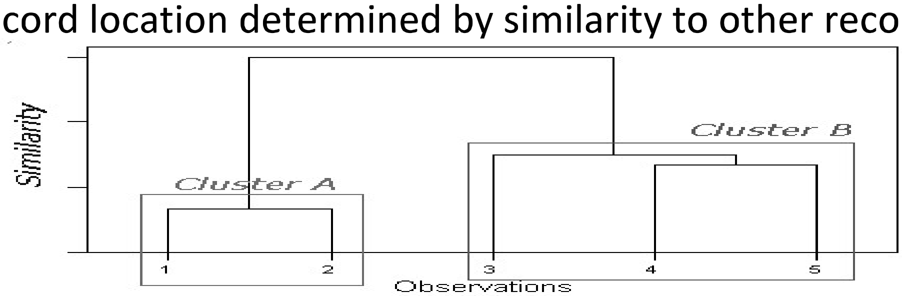
- Requires specifying number of clusters
- Computationally cheap

Hierarchical Clustering

Dendrogram:

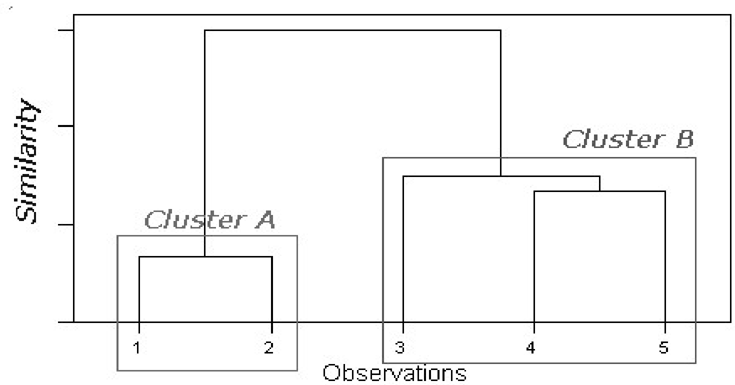
Tree-like diagram that summarize the clustering process

- Similar records joined by links.
- Record location determined by similarity to other records.



Hierarchical Clustering Algorithm

• Start with n clusters (1 record in each cluster)
• Step 1: two closest records are merged into one cluster



• At every step, pair of records/clusters with smallest distance are merged.
– Two records are merged,
– Or single record added to an existing cluster,
– Or two existing clusters are combined.
• Requires a definition of distance

Pairwise distance between Records

Distance Requirements: Non-negative ($d_{ij} > 0$)
$d_{ii} = 0$
Symmetry ($d_{ij} = d_{ji}$)
Triangle inequality ($d_{ij} + d_{jk} \geq d_{ik}$)

UG Business Programs
Universities Clustering.xls

• Data for 25 undergraduate programs at business schools in US universities in 1995.
• This dataset excludes image variables (student satisfaction, employer satisfaction and deans' opinions)

Univ	SAT	Top10	Accept	SFRatio	Expenses	GradRate
Brown	1310	89	22	13	22704	94
CalTech	1415	100	25	6	63575	81
CMU	1260	62	59	9	25026	72
Columbia	1310	76	24	12	31510	88
Cornell	1280	83	33	13	21864	90
Dartmout	1340	89	23	10	32162	95
Duke	1315	90	30	12	31585	95
Georgetov	1255	74	24	12	20126	92
Harvard	1400	91	14	11	39525	97
JohnsHop	1305	75	44	7	58691	87
MIT	1380	94	30	10	34870	91
Northwes	1260	85	39	11	28052	89
NotreDam	1255	81	42	13	15122	94
PennState	1081	38	54	18	10185	80
Princeton	1375	91	14	8	30220	95
Purdue	1005	28	90	19	9066	69
Stanford	1360	90	20	12	36450	93

Distance between two universities

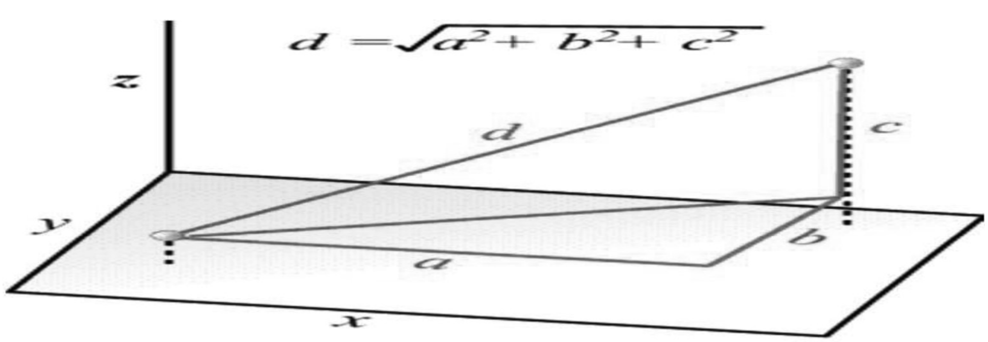
Notation:

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$$
$$x_J = (x_{J1}, x_{J2}, \ldots, x_{Jp})$$

Example:

• Caltech= (1415, 100, 25, 6, 63575, 81)
• Cornell = (1280, 83, 33, 13, 21864, 90)

Euclidean Distance



d_{iJ} = sqrt($(x_{i1}-x_{J1})^2 + (x_{i2}-x_{J2})^2 \ldots (x_{ip}-x_{Jp})^2$)

• 6-dimensional Euclidean distance between Caltech and Cornell:
Sqrt [(1415-1280)2 + (100-83)2 + (25-33)2 + (6-13)2 + (63575-21864)2 + (81-90)2] = 41,711.22

Standardize when there are multiple variables:

The units of the different measurements influence Euclidean distance. Solution: standardize (=normalize) each variable before measuring distances.

Standardizing Example:

Z_SAT = (SAT-Mean (SAT))/Stdev (SAT)

Univ	Z_SAT	Z_Top10	Z_Accept	Z_SFRatio	Z_Expenses	Z_GradRate
Brown	0.401994	0.644235	-0.87189	0.068840897	-0.32471667	0.80372917
CalTech	1.370988	1.210256	-0.71981	-1.65218153	2.508651168	-0.631501491
CMU	-0.05943	-0.74509	1.003685	-0.91460049	-0.16374483	-1.625122718
Columbia	0.401994	-0.0247	-0.77051	-0.17701945	0.285756214	0.141315019
Cornell	0.125139	0.335496	-0.31429	0.068840897	-0.38294938	0.362119736
Dartmouth	0.67885	0.644235	-0.8212	-0.66874014	0.330955887	0.914131529

Euclidean distance between Standardized Caltech and Cornell:
Sqrt [(1.371-1.125)2 + (1.210-0.335)2 + … + (-0.632-0.362)2]
= 3.84

Lots of other distance metrics

Statistical (Mahalanobis) distance:

→Uses correlation matrix

Manhattan distance

$D_{iJ} = | x_{i1}-x_{J1}|+| x_{i2}-x_{J2}|+\ldots..+| x_{ip}-x_{Jp}|$

Matching-type metrics for categorical data (next slide)

Distances for Binary Data

Similarity-based metrics based on 2x2 table of counts

	Married?	Smoker?	Manager?
Carries	Y	Y	Y
Sam	N	Y	N
Miranda	N	N	Y

		Miranda	
		N	Y
Carrie	N	0	0
	Y	2	1

		Miranda	
		N	Y
Carrie	N	a	b
	Y	c	d

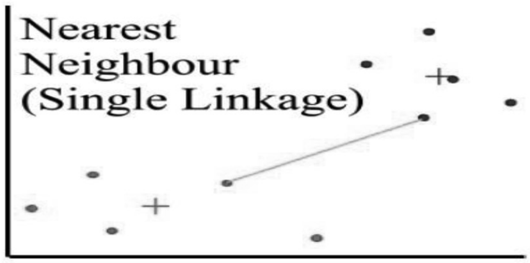
• Binary Euclidean Distance: (b+c)/ (a+b+c+d)
• Simple matching Coefficient: (a+d)/ (a+b+c+d)
• Jaquard's coefficient: d/ (b+c+d)
• For >2 categories, distance =0 only if both items have same category. Otherwise =1.

Distances for Mixed (numerical + Categorical) Data:

• Simple: standardize numerical variables to [0,1], then use Euclidian distance for all.

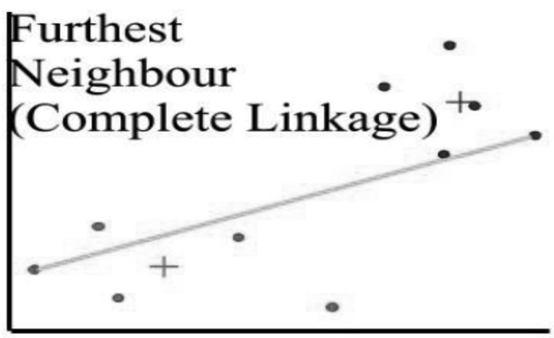
Distances Between Clusters: 'single linkage' ('nearest neighbor')

- Distance between 2 clusters = minimum distance between members of the two clusters



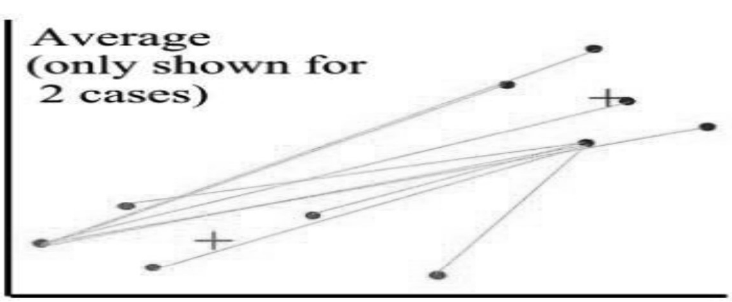
Distances Between Clusters: ‘complete linkage’ (‘farthest neighbor’)

- Distance between 2 clusters = greatest distance between members of the two clusters.



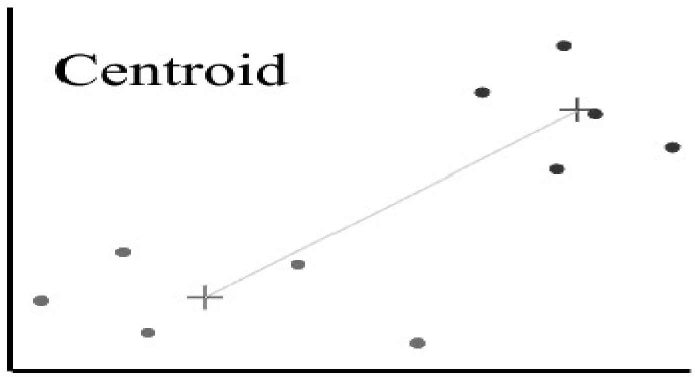
Distances Between Clusters: ‘average linkage’

• Distance between 2 clusters = average of all distances between members of the two clusters.



Distances Between Clusters: ‘centroid linkage’

• Distance between 2 clusters = distance between their centroids (Centers).



Pairwise distance between clusters

- Single linkage (nearest neighbor): minimum distance between members of the two clusters
- Complete linkage (farthest neighbor): greatest distance between members of the two clusters
- Average linkage: average of all distances between members of the two clusters.

. Centroid linkage: distance between their centroids (centers)

Once again: The Hierarchical Clustering Algorithm

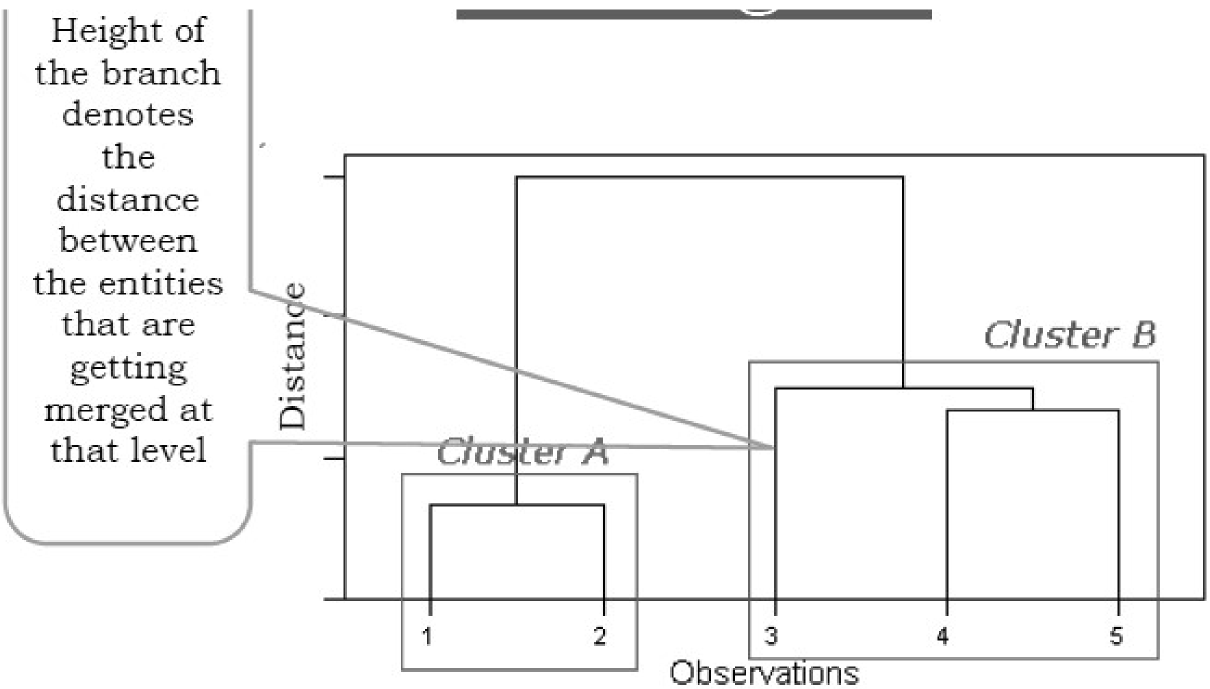
• Start with n clusters (record= cluster)

• Step 1: two closest records are merged into one cluster

• At every step, pair of clusters with smallest distance are merged

– Two records are merged, or single record added to an existing cluster, or two existing clusters are combined.

Hierarchical Clustering: The Dendrogram



Why cluster universities?

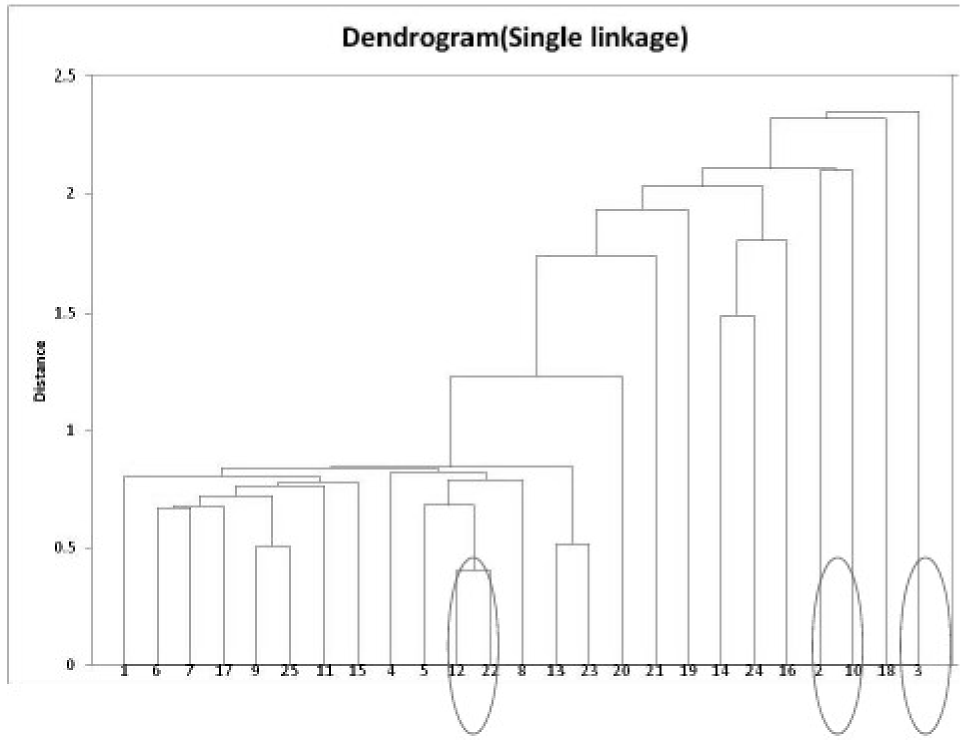
- How can clustering help a prospective applicant?
- How can clustering help a business school dean?

Evaluating usefulness of clustering

- What characterizes each cluster?
- Can you give a “name” to each cluster?
- Does this give us any insight?

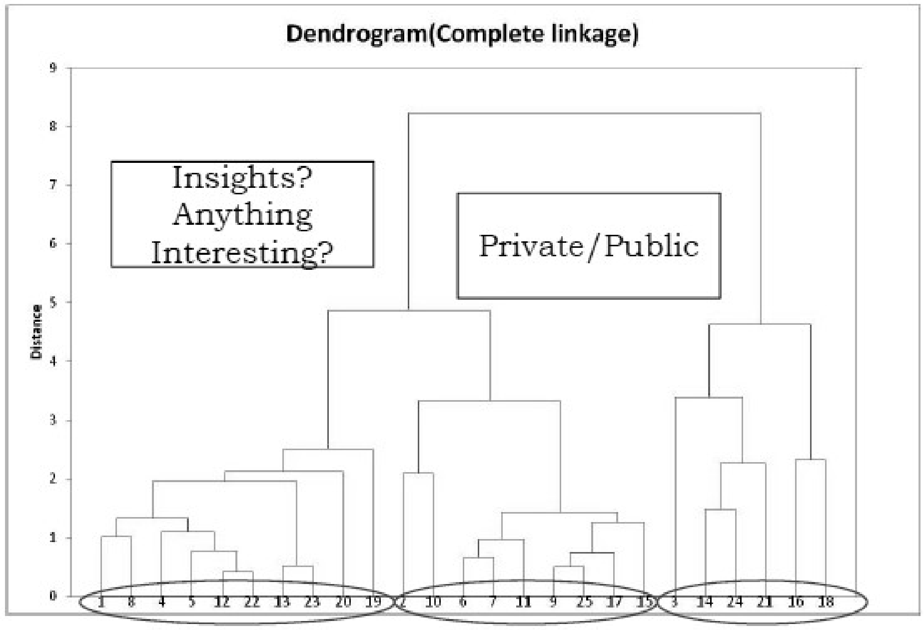
Insights? Anything Interesting?

Row Id.	University
1	Brown
2	CalTech
3	CMU
4	Columbia
5	Cornell
6	Dartmouth
7	Duke
8	Georgetown
9	Harvard
10	JohnsHopkins
11	MIT
12	Northwestern
13	NotreDame
14	PennState
15	Princeton
16	Purdue
17	Stanford
18	TexasA&M
19	UCBerkeley
20	UChicago
21	UMichigan
22	UPenn
23	UVA
24	UWisconsin
25	Yale



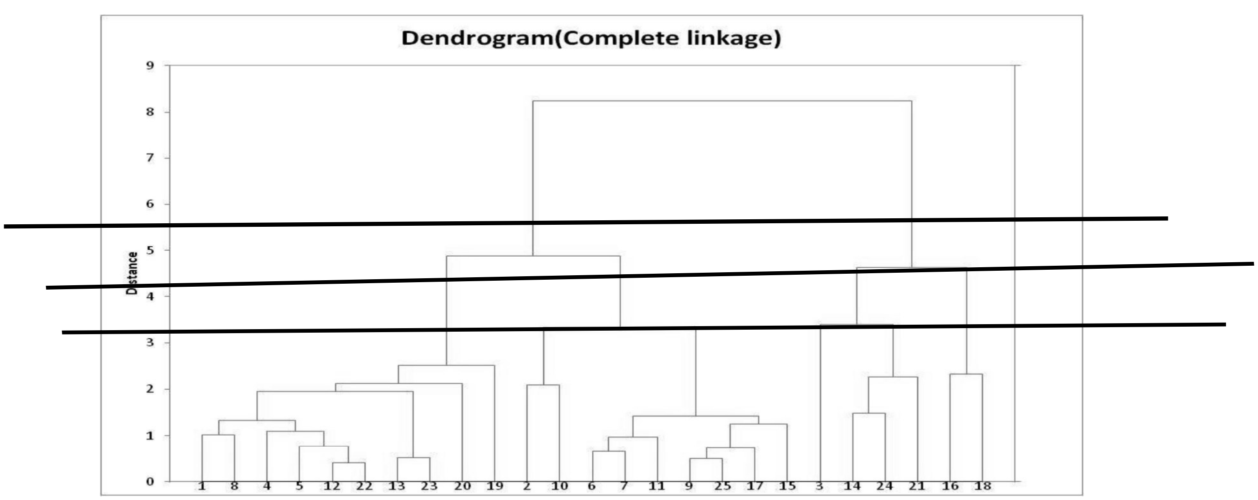
Dendrogram for Business School Example with Complete Linkage

Row Id.	University	Cluster Id
1	Brown	1
2	CalTech	1
3	CMU	2
4	Columbia	1
5	Cornell	1
6	Dartmouth	1
7	Duke	1
8	Georgetown	1
9	Harvard	1
10	JohnsHopkins	1
11	MIT	1
12	Northwestern	1
13	NotreDame	1
14	PennState	2
15	Princeton	1
16	Purdue	2
17	Stanford	1
18	TexasA&M	2
19	UCBerkeley	1
20	UChicago	1
21	UMichigan	2
22	UPenn	1
23	UVA	1
24	UWisconsin	2
25	Yale	1

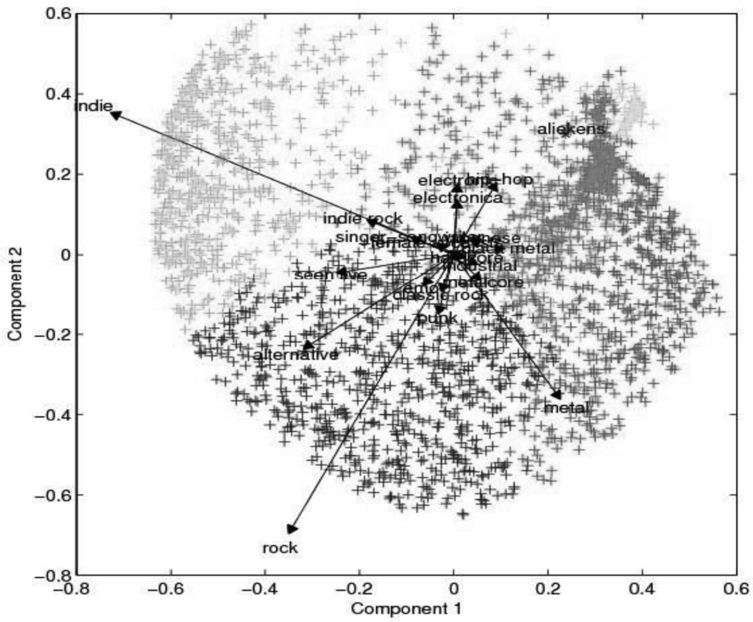


From Dendrograms to Clusters

- After dendrogram is obtained, cut it to create clusters. How?
- Examine distance levels
- Cut point determines # clusters
- Obtain statistics on resulting clusters



Non-Hierarchical Clustering:



K-means clustering

• Predetermined number (K) of non-overlapping clusters
• Clusters are homogeneous yet dissimilar to other clusters
• Need measures of within-cluster similarity (homogeneity) and between-cluster similarity
• No hierarchy (no dendrogram)! End Product is final cluster memberships.
. Useful for large data sets.

K-means clustering

Algorithm minimizes within-cluster variance (heterogeneity)

1. For a user-specified value of K, partition dataset into K initial clusters (next slide).
2. For each record, assign it to cluster with closest centroid

3. Re-calculate centroids for the “losing” and “receiving” clusters. can be done
• After reassignment of each record, or
• After one complete pass through all records (cheaper)
4. Repeat Steps 2-3 until no more reassignment is necessary

Initial partition into K clusters

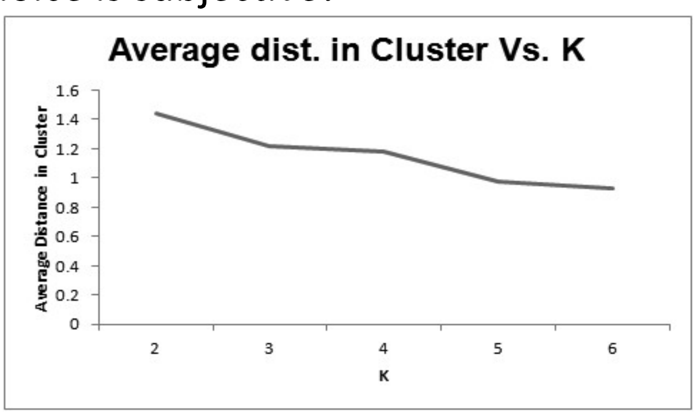
Initial partitions can be obtained by either
1. user-specified initial partitions, or
2. user-specified initial centroids, or
3. Random partitions

Stability: Run algorithm with different initial partitions

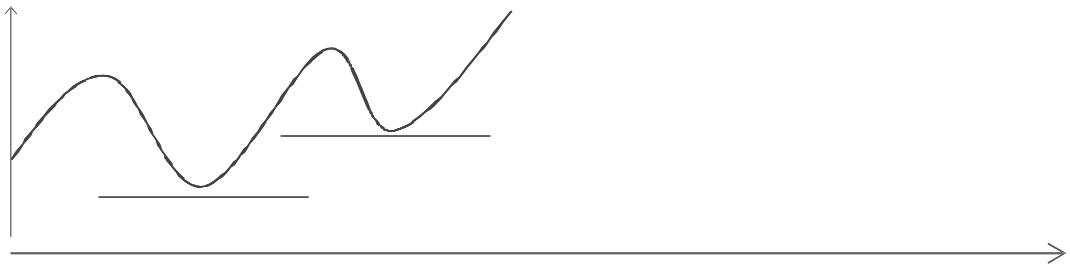
Selecting K

• Re-run algorithm for different values of K
• Tradeoff: simplicity (interpretation) vs. adequacy (within-cluster homogeneity)
• Plot within-cluster variability as a function of K

Choice is subjective!



Why multiple start points (initial partitions) may be necessary?
• K-means clustering is a minimization problem
• Existence of multiple local minima



Convergence/robustness of K-means

• Procedure might oscillate indefinitely
• Convergence criterion:
– Stop when a cluster centroid moves less than a % of smallest distance between any of the centroids
– Specify the maximum number of iterations

Final checks

• Cluster stability: do cluster assignments change dramatically if some inputs are slightly altered?
• Cluster separation: compare between-cluster variation to within cluster variation.

K-Means vs. Hierarchical

K-Means

The Good
• Computationally fast for large datasets
• Useful when certain K needed

The Bad
• Can take long to terminate
• Final solution not guaranteed to be "globally optimal"
• Different initial partitions can lead to different solutions
• Must re-run the algorithm for different values of K
• No dendrogram

Hierarchical

The Good

• Finds “natural” grouping – no need to specify number of clusters

• Dendrogram: transparency of process, good for presentation

The Bad

• Require computation & storage of n x n distance matrix

• Algorithm makes only one pass through the data. Records that are incorrectly allocated early on cannot be reallocated subsequently

• Low stability: Reordering data or dropping a few records can lead to different solution

• Single complete linkage robust to distance metric as long as the relative ordering is kept. Average linkage is
NOT.

• Most distances sensitive to outliers.