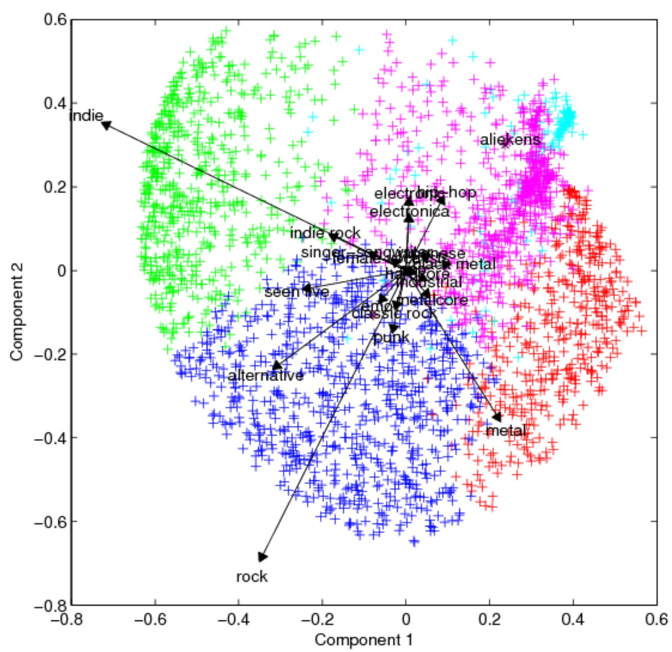
Non-Hierarchical Clustering: K-Means Clustering



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! End-product is final cluster memberships (no dendrogram)
- Useful for large datasets

K-means clustering

Iterative procedure:

- Start from K initial clusters
- Each record reassigned to cluster with "closest" centroid
- Stop when further reassignments make clusters less homogenous

Algorithm minimizes within-cluster *variance* (heterogeneity)

K-means algorithm

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 nearest 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 reassignments occur

Initial partition into K clusters

Initial partitions can be obtained by either

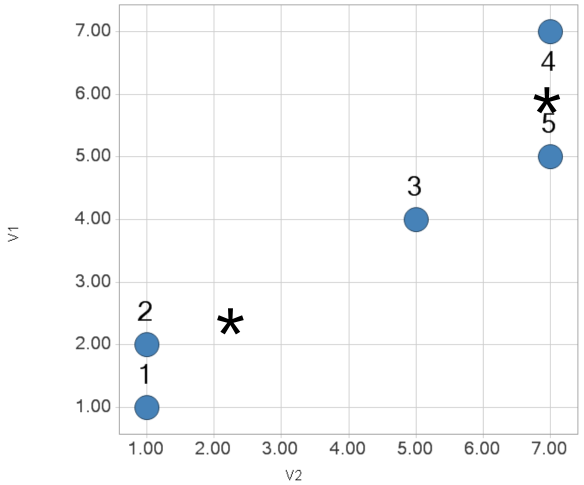
1. user-specified initial partitions, or
2. user-specified initial centroids, or
3. random partitions.

Info from external variable

Stability: run algorithm with different initial partitions

Example: K=2

item	v1	v2
1	1	1
2	2	1
3	4	5
4	7	7
5	5	7



Start with cluster A: 1,2,3 and cluster B: 4,5

Compute cluster centroids (next slide)

What are the centroids of clusters A and B?

item	v1	v2
1	1	1
2	2	1
3	4	5
4	7	7
5	5	7

1. A= (1,1.5,4.5) and B=(7,6)
2. A= (2.33) and B=(6.5)
3. A= (2.33,2.33) and B=(6,7)

Example – cont.

Compute Euclidean distance of each record from each centroid, and re-assign to closest cluster.

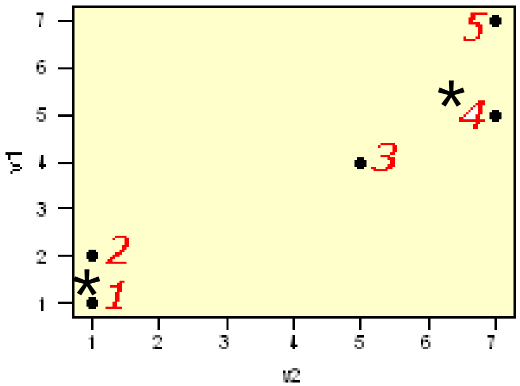
	Cluster A	**Cluster B**
Item 1	$\sqrt{(1-2.33)^2 + (1-2.33)^2} = 1.89$	$\sqrt{(1-6)^2 + (1-7)^2} = 7.81$
Item 2	1.37	7.21
Item 3	$\sqrt{(4-2.33)^2 + (5-2.33)^2} = 3.14$	$\sqrt{(4-6)^2 + (5-7)^2} = 2.83$
Item 4	6.60	1
Item 5	5.37	1

First iteration results

Cluster A: 1,2   Cluster B: 3,4,5

Re-compute centroids:

cent(A)= (__1.5__, __1__)   cent(B)= (__5.33__, __6.33__)



Re-compute distances of records to centroids

	Cluster A	Cluster B
Item 1	$\sqrt{(1-1.5)^2 + (1-1)^2} = 0.5$	$\sqrt{(1-5.33)^2+(1-6.33)^2} = 6.87$
Item 2	0.5	6.29
Item 3	$\sqrt{(4-1.5)^2+ (5-1)^2} = 4.72$	$\sqrt{(4-5.33)^2+ (5-6.33)^2} =1.89$
Item 4	8.14	1.80
Item 5	6.95	0.75

Stop here!

Using XLMiner: Universities Example

K=3

CMU
PennState
Purdue
TexasA&M
UMichigan
UWisconsin

Brown
Columbia
Cornell
Duke
Georgetown
Northwestern
NotreDame
UCBerkeley
UChicago
UPenn
UVA

CalTech
Dartmouth
Harvard
JohnsHopkins
MIT
Princeton
Stanford
Yale

Cluster	SAT	Top10	Accept	SFRatio	Expenses	GradRate
Cluster-1	1114.33383	46.9999397	67.8333949	16.9999956	13384.6711	73.999965
Cluster-2	1275.00006	82.2727323	34.9090981	12.8181815	24125.9028	89.9090876
Cluster-3	1368.7501	90.6249908	23.6250019	9.37500161	42375.8079	91.8750008

Evaluating usefulness of clustering

What characterizes each cluster?

Can you give a “name” to each cluster?

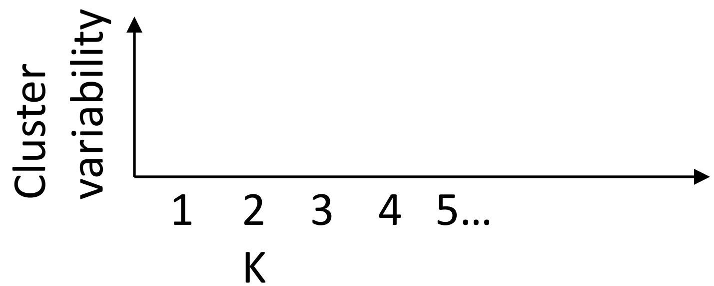
Does this give us any insight?

Selecting K

Re-run algorithm for different values of *K*

Tradeoff: simplicity (interpretation) vs. adequacy (within-cluster homogeneity)

Elbow graph: within-cluster variability as a function of K



Choice is subjective!

Worksheets
KM_Output1, KM_Output2

Universities example: K=2 vs. K=3

Inter cluster distance	Cluster-1	Cluster-2
Cluster-1	0	18426.60302
Cluster-2	18426.60302	0

ary

Cluster	#Obs	Average distance in cluster
Cluster-1	6	1.769
Cluster-2	19	1.338
Overall	25	1.442

Inter cluster distance	Cluster-1	Cluster-2	Cluster-3
Cluster-1	0	10742.55419	28992.3261
Cluster-2	10742.55419	0	18250.15169
Cluster-3	28992.3261	18250.15169	0

ary

Cluster	#Obs	Average distance in cluster
Cluster-1	6	1.769
Cluster-2	11	0.953
Cluster-3	8	1.172
Overall	25	1.219

Worksheets
KM_Output1- KM_Output5

“Elbow” chart for choosing K

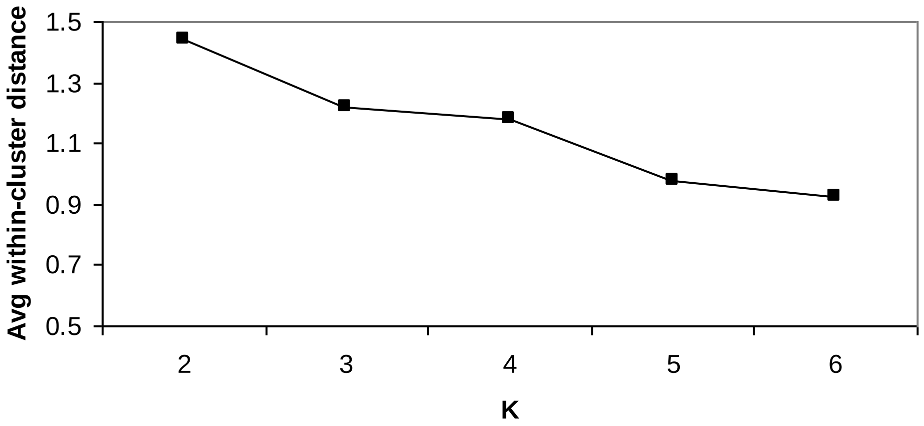
Cluster	#Obs	Average distance in cluster
Cluster-1	6	1.769
Cluster-2	19	1.338
Overall	25	1.442

Cluster	#Obs	Average distance in cluster
Cluster-1	6	1.769
Cluster-2	11	0.953
Cluster-3	8	1.172
Overall	25	1.219

Cluster	#Obs	Average distance in cluster
Cluster-1	4	1.563
Cluster-2	11	0.953
Cluster-3	2	1.668
Cluster-4	8	1.172
Overall	25	1.178

Cluster	#Obs	Average distance in cluster
Cluster-1	3	1.296
Cluster-2	7	0.629
Cluster-3	3	1.527
Cluster-4	10	0.952
Cluster-5	2	1.049
Overall	25	0.98

Cluster	#Obs	Average distance in cluster
Cluster-1	3	1.296
Cluster-2	5	0.898
Cluster-3	6	0.598
Cluster-4	3	1.527
Cluster-5	6	0.753
Cluster-6	2	1.049
Overall	25	0.926



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.

http://www.clustan.com/k-means_critique.html (some interesting points about outliers, different starting points, and more)

K-means: Advantages and Disadvantages of

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