Methods for Intelligent Systems

Lecture Notes on Clustering (II)
2009-2010

Davide Eynard

eynard@elet.polimi.it

Department of Electronics and Information

Politecnico di Milano

Course Schedule [*Tentative*]

Date	Topic
11/03/2010	Clustering: Introduction
18/03/2010	Clustering: K-means & Hierarchical
25/03/2010	Clustering: Fuzzy, Gaussian & SOM
08/04/2010	Clustering: PDDP & Vector Space Model
15/04/2010	Clustering: Limits, DBSCAN & Jarvis-Patrick
29/04/2010	Clustering: Evaluation Measures

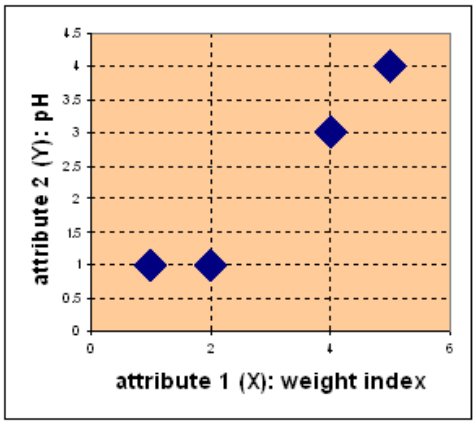
K-Means Algorithm

- One of the simplest unsupervised learning algorithms
- Assumes Euclidean space *(works with numeric data only)*
- Number of clusters fixed a priori
- **How does it work?**

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group *centroids*.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move.

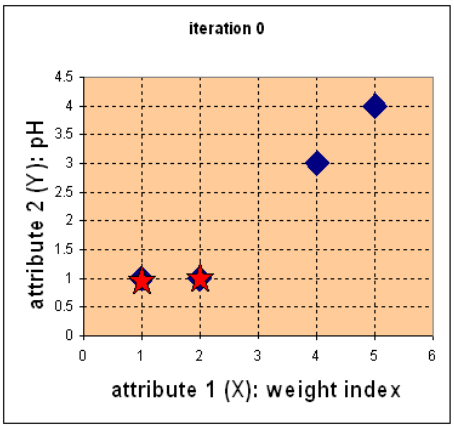
K-Means: A numerical example

Object	Attribute 1 (X)	Attribute 2 (Y)
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4



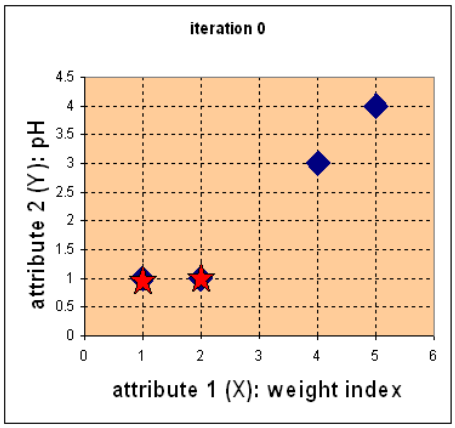
K-Means: A numerical example

- Set initial value of centroids
  - $c_1 = (1, 1)$, $c_2 = (2, 1)$



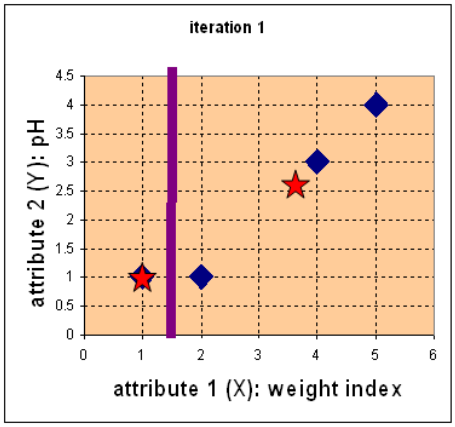
K-Means: A numerical example

- Calculate Objects-Centroids distance
  - $D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \begin{matrix} c_1 = (1, 1) \\ c_2 = (2, 1) \end{matrix}$



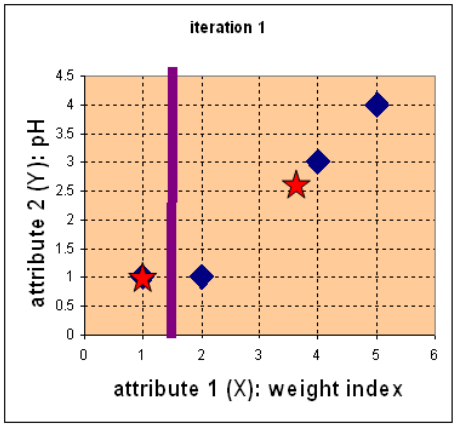
K-Means: A numerical example

- Object Clustering
  - $G^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} group1 \\ group2 \end{matrix}$



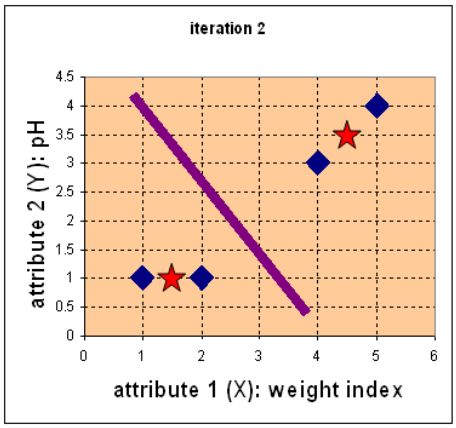
K-Means: A numerical example

- Determine new centroids
  - $c_1 = (1, 1)$
    $c_2 = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = \left(\frac{11}{3}, \frac{8}{3}\right)$



K-Means: A numerical example

- $D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \begin{matrix} c_1 = (1,1) \\ c_2 = (\frac{11}{3}, \frac{8}{3}) \end{matrix}$
- $G^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \Rightarrow \begin{matrix} c_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1.5, 1) \\ c_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4.5, 3.5) \end{matrix}$



K-Means: still alive?

Time for some demos!

K-Means: Summary

- Advantages:
  - Simple, understandable
  - Relatively efficient: $O(tkn)$, where n is #objects, k is #clusters, and t is #iterations $(k, t \ll n)$
  - Often terminates at a local optimum
- Disadvantages:
  - Works only when mean is defined (what about categorical data?)
  - Need to specify k, the number of clusters, in advance
  - Unable to handle noisy data (too sensible to outliers)
  - Not suitable to discover clusters with non-convex shapes
  - Results depend on the metric used to measure distances and on the value of k
- Suggestions
  - Choose a way to initialize means (i.e. randomly choose k samples)
  - Start with *distant* means, run many times with different starting points
  - Use another algorithm ;-)

K-Medoids

- K-Means algorithm is too sensitive to outliers
  - An object with an extremely large value may substantially distort the distribution of the data
- **Medoid**: the most centrally located point in a cluster, as a representative point of the cluster
- Note: while a medoid is always a point inside a cluster too, a centroid could be not part of the cluster.
- Instead of *means*, use *medians* of each cluster
  - Mean of 1, 3, 5, 7, 9 is 5
  - Mean of 1, 3, 5, 7, 1009 is 205
  - Median of 1, 3, 5, 7, 1009 is 5

PAM

PAM means **P**artitioning **A**round **M**edoids. The algorithm follows:

1. Given k
2. Randomly pick k instances as initial medoids
3. Assign each data point to the nearest medoid x
4. Calculate the objective function
   - the sum of dissimilarities of all points to their nearest medoids. (squared-error criterion)
5. Randomly select a point y
6. Swap x by y if the swap reduces the objective function
7. Repeat (3-6) until no change

PAM

- Pam is more robust than k-means in the presence of noise and outliers
  - A medoid is less influenced by outliers or other extreme values than a mean (why?)
- Pam works well for small data sets but does not scale well for large data sets
  - $O(k(n-k)^2)$ for each change where n is # of data objects, k is # of clusters

Hierarchical Clustering

- Top-down vs Bottom-up
- Top-down (or *divisive*):
  - Start with one universal cluster
  - Split it into two clusters
  - Proceed recursively on each subset
  - (can be very fast)
- Bottom-up (or *agglomerative*):
  - Start with single-instance clusters ("every item is a cluster")
  - At each step, join the two closest clusters
  - (design decision: distance between clusters)

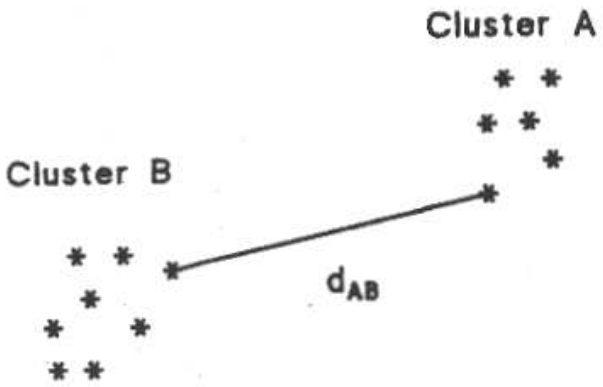
Hierarchical Clustering Algorithm

Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering is the following:

1. Start by assigning each item to a cluster. Let the distances (similarities) between the clusters be the same as the distances (similarities) between the items they contain.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster. Now, you have one cluster less.
3. Compute distances (similarities) between the new cluster and each of the old ones.
4. Repeat Steps 2 and 3 until all items are clustered into a single cluster of size N.

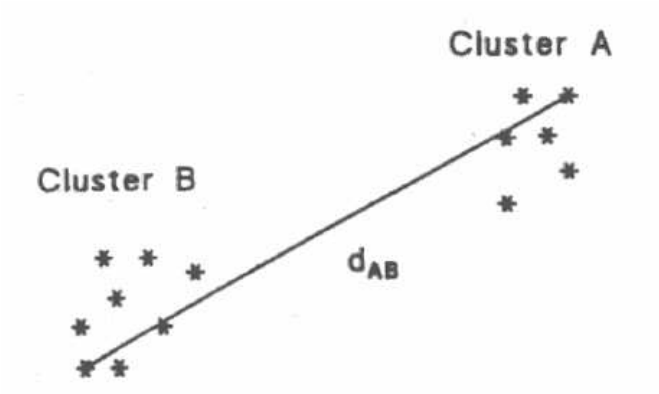
Single linkage clustering

- We consider the distance between two clusters to be equal to the **shortest** distance from any member of one cluster to any member of the other one (**greatest** similarity).



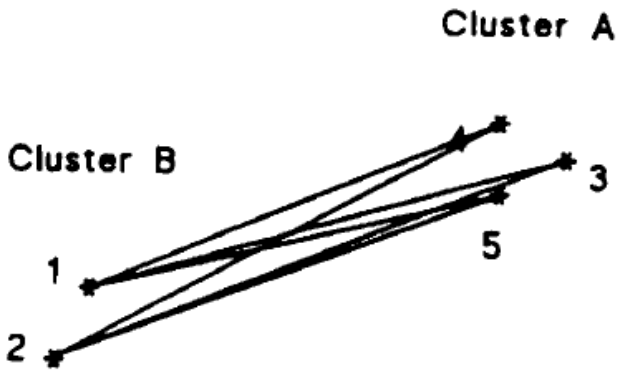
Complete linkage clustering

- We consider the distance between two clusters to be equal to the **greatest** distance from any member of one cluster to any member of the other one (**smallest** similarity).



Average linkage clustering

- We consider the distance between two clusters to be equal to the **average** distance from any member of one cluster to any member of the other one.



About distances

If the data exhibit strong clustering tendency, all 3 methods produce similar results.

- **SL**: requires only a single dissimilarity to be small. Drawback: produced clusters can violate the “compactness” property (cluster with large diameters)
- **CL**: opposite extreme (compact clusters with small diameters, but can violate the “closeness” property)
- **AL**: compromise, it attempts to produce relatively compact clusters and relatively far apart. BUT it depends on the dissimilarity scale.

Hierarchical clustering: Summary

- Advantages
  - It's nice that you get a hierarchy instead of an amorphous collection of groups
  - If you want k groups, just cut the $(k-1)$ longest links
- Disadvantages
  - It doesn't scale well: time complexity of at least $O(n^2)$, where n is the number of objects
  - It cannot undo what was done previously
  - It has no real statistical or information-theoretic foundations

Hierarchical Clustering Demo

Time for another demo!

Bibliography

- A Tutorial on Clustering Algorithms Online tutorial by M. Matteucci
- K-means and Hierarchical Clustering
  Tutorial Slides by A. Moore
- "Metodologie per Sistemi Intelligenti" course - Clustering
  Tutorial Slides by P.L. Lanzi
- K-Means Clustering Tutorials
  Online tutorials by K. Teknomo
- As usual, more info on del.icio.us

- The end