Cluster Algorithms

Adriano Cruz
adriano@nce.ufrj.br

28 de outubro de 2013

Summary

1. K-Means

Summary

1. K-Means
2. Fuzzy C-means

Summary

Summary

Summary

Summary

Section Summary

K-Means Algorithm

1. Based on the Euclidean distances among elements of the cluster.
2. Centre of the cluster is the mean value of the objects in the cluster.
3. Classifies objects in a hard way.
4. Each object belongs to a single cluster.

Initial Definitions

1. Consider n objects and c clusters.
2. Each object $\mathbf{x}_e \in X$ is defined by l characteristics $\mathbf{x}_e = (x_{e,1}, x_{e,2}, ..., x_{e,l})$.
3. Consider A a set of c clusters $(A = A_1, A_2, ..., A_c)$.

K-Means Properties

1. The union of all c clusters makes the Universe

$$\cup_{i \in c} A_i = X$$

.

2. No element belongs to more than one cluster.

$$\forall i, j \in c : i \neq j \Rightarrow A_i \cap A_j = \emptyset$$

3. There is no empty cluster

$$\emptyset \neq A_i \neq X$$

.

Membership Function

$$\chi_{A_i}(x_e) = \begin{cases} 1 & x_e \in A_i \\ 0 & x_e \notin A_i \end{cases}$$

$$\sum_{i=1}^{c} \chi_{A_i}(x_e) \equiv \sum_{i=1}^{c} \chi_{ie} = 1, \forall e$$

$$\chi_{ie} \times \chi_{je} = 0, \forall e$$

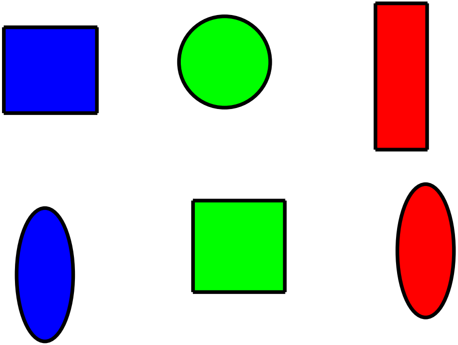
$$0 < \sum_{e=1}^{n} \chi_{ie} < n$$

Membership Matrix U

1. Matrix containing the values of inclusion of each element into each cluster (0 or 1).
2. Matrix has c (clusters) lines and n (elements) columns.
3. The sum of all elements in the column must be equal to one (element belongs only to one cluster
4. The sum of each line must be less than n e grater than 0. No empty cluster, or cluster containing all elements.

Matrix Examples

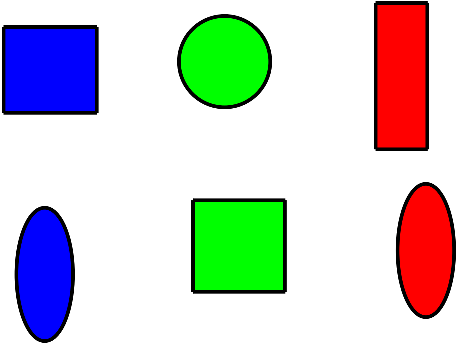
1. Two examples of clustering.
2. What do the matrices represent?



1	0	1	0	1	0
0	1	0	1	0	1

1	0	0	1	0	0
0	1	0	0	1	0
0	0	1	0	0	1

1. Matrices U_1 and U_2 represent the same clusters.



U_1

1	0	1	0	1	0
0	1	0	1	0	1

U_2

0	1	0	1	0	1
1	0	1	0	1	0

K-Means inputs and outputs

1. Inputs: the number of clusters c and a database containing n objects with l characteristics each.
2. Output: A set of c clusters that minimises the square-error criterion.

K-Means Algorithm v1

Arbitrarily assigns each object to a cluster (matrix U).
repeat
  Update the cluster centres;
  Reassign objects to the clusters to which the objects are most similar;
until no change;

K-Means Algorithm v2

Arbitrarily choose c objects as the initial cluster centres.
repeat
  Reassign objects to the clusters to which the objects are most similar;
  Update the cluster centres;
until no change;

Algorithm details

1. The algorithm tries to minimize the function

$$J(U, v) = \sum_{e=1}^{n} \sum_{i=1}^{c} \chi_{ie} (d_{ie})^2$$

2. $(d_{ie})^2$ is the distance between the element $\mathbf{x_e}$ (m characteristics) and the centre of the cluster i ($\mathbf{v_i}$)

$$\begin{aligned} d_{ie} &= d(\mathbf{x_e} - \mathbf{v_i}) \\ d_{ie} &= ||\mathbf{x_e} - \mathbf{v_i}|| \\ d_{ie} &= \left[\sum_{j=1}^{l} (x_{ej} - v_{ij})^2 \right]^{1/2} \end{aligned}$$

Cluster Centres

1. The centre of the cluster i ($\mathbf{v_i}$) is an l characteristics vector.
2. The jth co-ordinate is calculated as

$$v_{ij} = \frac{\displaystyle\sum_{e=1}^{n} \chi_{ie} \cdot x_{ej}}{\displaystyle\sum_{e=1}^{n} \chi_{ie}}$$

Detailed Algorithm

Choose c (number of clusters).
Set error ($\varepsilon > 0$) and step ($r = 0$).
Arbitrarily set matrix $U(r)$. Do not forget, each element belongs to a single cluster, no empty cluster and no cluster has all elements.

repeat

Calculate the centre of the clusters v_i^r

Calculate the distance d_i^r of each point to the centre of the clusters

Generate U^{r+1} recalculating all characteristic functions using the equation

$$\chi_{ie}^{r+1} = \begin{cases} 1 & d_{ie}^r = \min(d_{je}^r), \forall j \in k \\ 0 & \end{cases}$$

until $||U^{r+1} - U^r|| < \varepsilon$

Matrix Norms

1. Consider a matrix U of n lines and n columns:
2. Column norm $= ||A||_1 = \max\limits_{1 \leq j \leq n} \sum\limits_{i=1}^{n} |a_{ij}|$
3. Line norm $= ||A||_\infty = \max\limits_{1 \leq i \leq n} \sum\limits_{j=1}^{n} |a_{ij}|$

Stop criteria

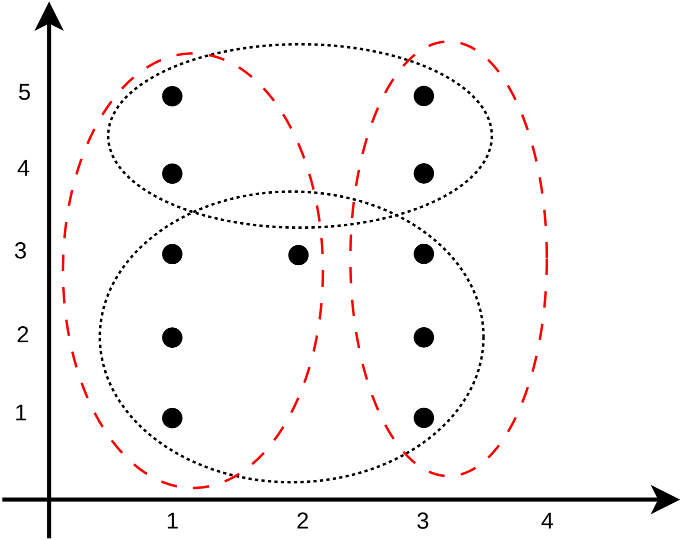
1. Some implementatios use the value of the objective function to stop the algorithm
2. The algorithm will stop when the objective function improvement between two consecutive iterations is less than the minimum amount of improvement specified.
3. The objective function is

$$J(U, v) = \sum_{e=1}^{n} \sum_{i=1}^{c} \chi_{ie} (d_{ie})^2$$

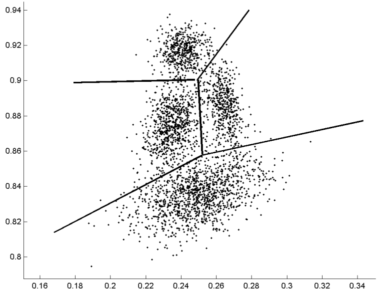
K-Means Problems

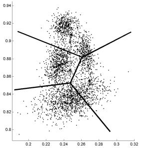
1. Suitable when clusters are compact clouds well separated.
2. Scalable because computational complexity is $O(nkr)$.
3. Necessity of choosing c is disadvantage.
4. Not suitable for non convex shapes.
5. It is sensitive to noise and outliers because they influence the means.
6. Depends on the initial allocation.

Examples of Result



Original x Result Data





Section Summary

Fuzzy C-means

1. Fuzzy version of K-means
2. Elements may belong to more than one cluster
3. Values of characteristic function range from 0 to 1.
4. It is interpreted as the degree of membership of an element to a cluster relative to all other clusters.

Initial Definitions

1. Consider n objects and c clusters.
2. Each object $x_e \in X$ is defined by l characteristics $x_i = (x_{e1}, x_{e2}, ..., x_{el})$.
3. Consider A a set of c clusters $(A = A_1, A_2, ..., A_c)$.

C-Means Properties

1. The union of all c clusters makes the Universe

$$\cup_{i \in c} A_i = X$$

.

2. There is no empty cluster

$$\emptyset \neq A_i \neq X$$

.

Membership Function

$$\mu_{A_i}(x_e) \equiv \mu_{ie} \in [0..1]$$

$$\sum_{i=1}^{c} \mu_{A_i}(x_e) \equiv \sum_{i=1}^{c} \mu_{ie} = 1, \forall e$$

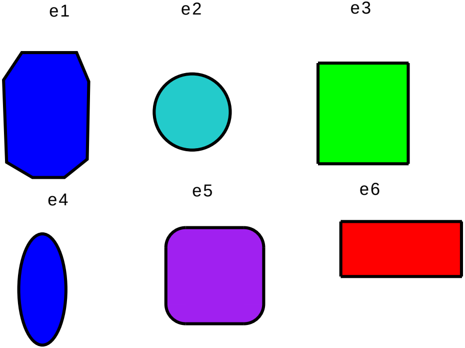
$$0 < \sum_{e=1}^{n} \mu_{ie} < n$$

Membership Matrix

1. Matrix containing the values of inclusion of each element into each cluster [0,1].
2. Matrix has c (clusters) lines and n (elements) columns.
3. The sum of all elements in the column must be equal to one.
4. The sum of each line must be less than n e grater than 0.
5. No empty cluster, or cluster containing all elements.

Matrix Examples

1. Two examples of clustering.
2. What do the clusters represent?



0.9	0	1	0	0.85	0.95
0.1	1	0	1	0.15	0.05

0	0.2	0	0	0.4	1
0	0.4	1	0	0.1	0
1	0.4	0	1	0.5	0

C-Means Algorithm v1

Arbitrarily assigns each object to a cluster (matrix U).
repeat
  Update the cluster centres;
  Reassign objects to the clusters to which the objects are most similar;
until no change;

C-Means Algorithm v2

Arbitrarily choose c objects as the initial cluster centres.

repeat

  Reassign objects to the clusters to which the objects are most similar;

  Update the cluster centres;

until no change;

Algorithm details

1. The algorithm tries to minimize the function

$$J(U, v) = \sum_{e=1}^{n} \sum_{i=1}^{c} \mu_{ie}^{m} (d_{ie})^2$$

2. m is the nebulization factor.
3. $(d_{ie})^2$ is the distance between the element $\mathbf{x_e}$ (m characteristics) and the centre of the cluster i ($\mathbf{v_i}$)

$$\begin{aligned} d_{ie} &= d(\mathbf{x_e} - \mathbf{v_i}) \\ d_{ie} &= ||\mathbf{x_e} - \mathbf{v_i}|| \\ d_{ie} &= \left[\sum_{j=1}^{l} (x_{ej} - v_{ij})^2 \right]^{1/2} \end{aligned}$$

Nebulization Factor

1. m is the nebulization factor.
2. This value has a range $[1, \infty)$
3. If $m = 1$ the the system is crisp.
4. If $m \to \infty$ then all the membership values tend to $1/c$.
5. The most common values are 1.25 and 2.0

Cluster Centres

1. The centre of the cluster i ($\mathbf{v_i}$) is an l characteristics vector.
2. The jth co-ordinate is calculated as

$$v_{ij} = \frac{\displaystyle\sum_{e=1}^{n} \mu_{ie}^{m} \cdot x_{ej}}{\displaystyle\sum_{e=1}^{n} \mu_{ie}^{m}}$$

Detailed Algorithm

Choose c (number of clusters).
Set error ($\varepsilon > 0$), nebulization factor (m) and step ($r = 0$).
Arbitrarily set matrix $U(r)$. Do not forget, each element belongs to a single cluster, no empty cluster and no cluster has all elements.

repeat

Calculate the centre of the clusters v_i^r

Calculate the distance d_i^r of each point to the centre of the clusters

Generate U^{r+1} recalculating all characteristic functions. How?

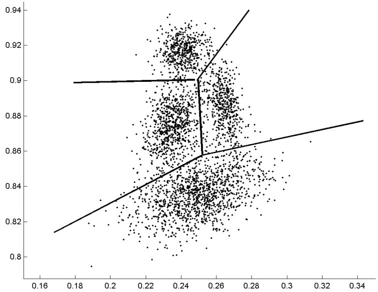
until $||U^{r+1} - U^r|| < \varepsilon$

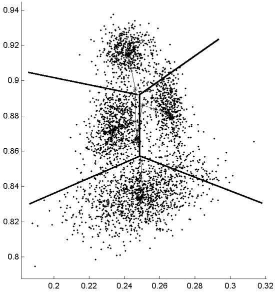
How to recalculate?

1. If there is any distance greater than zero then membership grade is the weighted average of the distances to all centers.
2. Otherwise the element belongs to this cluster and no other one.

$$
\begin{aligned}
\text{if } d_{ie} > 0, \quad & \forall \quad i \in [1..c] \\
\text{then } \mu_{ie} \quad &= \quad \left[\sum_{k=1}^{c} \left[\frac{d_{ie}}{d_{ke}} \right]^{\frac{2}{m-1}} \right]^{-1} \\
\text{else if } d_{ie} \quad &= \quad 0 \text{ then } \mu_{ie} = 1 \text{ else } \mu_{ie} = 0
\end{aligned}
$$

Original x Result Data





Section Summary

$$\mu_{A_i}(x_e) \equiv \mu_{ie} \in [0..1]$$
$$\sum_{i=1}^{c} \mu_{A_i}(x_e) \equiv \sum_{i=1}^{c} \mu_{ie} > 0, \forall e$$

1. The membership degree is the representativity of typicality of the datum x related to the cluster i.

Algorithm details - I

1. The algorithm tries to minimize the function

$$J(U, v) = \sum_{e=1}^{n} \sum_{i=1}^{c} \mu_{ie}^{m} (d_{ie})^2 + \sum_{i=1}^{c} \eta_i \sum_{e=1}^{n} (1 - \mu_{ie})^m$$

2. The first sum is the usual and the second rewards high memberships.

Algorithm details - II

1. It is possible to prove that in order to minimize the function J the membership degree must be calculated as:

$$\mu_{ie} = \frac{1}{1 + \left(\frac{d_{ie}^2}{\eta_i}\right)^{\frac{1}{m-1}}}$$

2. Where

$$\eta_i = \frac{\sum_{i=1}^{n} \mu_{ie}^m d_{ie}^2}{\sum_{i=1}^{n} \mu_{ie}^m}$$

3. η_i is a factor that controls the expansion of the cluster.

Algorithm details - III

1. η_i is a factor that controls the expansion of the cluster.
2. For instance, if η_i is equal to the distance d_{ie}^2 then at this distance the membership will be equal to 0.5.
3. So using η_i it is possible to control the membership degrees.
4. η_i is usually estimated.

$$\mu_{ie} = \frac{1}{1 + \left(\frac{d_{ie}^2}{\eta_i}\right)^{\frac{1}{m-1}}}$$

$$\eta_i = d_{ie}^2$$

$$\mu_{ie} = 0.5$$

Detailed Algorithm

1. It seems natural to repeat the algorithms K-means and C-Means that are similar.
2. However this does not give satisfactory results.
3. The algorithm tends to interpret data with low membership in all clusters as outliers instead of adjusting the results.
4. So a initial probabilistic clustering is performed.

Detailed Algorithm

Choose c (number of clusters).
Set error ($\varepsilon > 0$), nebulization factor (m) and step ($r = 0$).
Execute Algorithm Fuzzy C-Means

Detailed Algorithm cont.

for 2 TIMES **do**
  Initialize U^0 and Cluster Centres with the previous results
  $r \leftarrow 0$
  Initialize η_i with the previous results
  repeat
    $r \leftarrow r + 1$
    Calculate the Cluster Centres using U^{s-1}
    Calculate U^s
  until $||U^{r+1} - U^r|| < \varepsilon$
end for

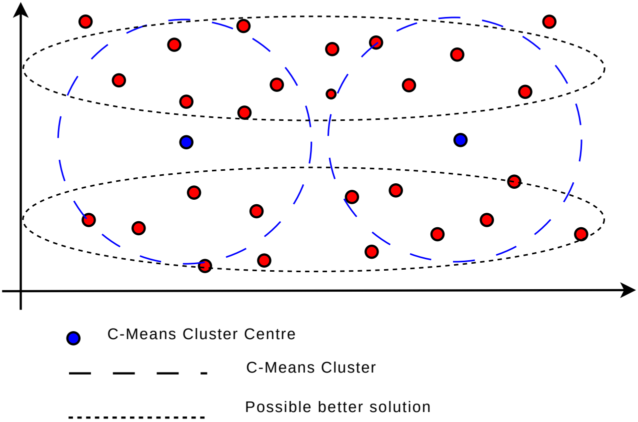
Section Summary

GK Method

1. This method (GK) is similar to the Fuzzy C-means (FCM).
2. The difference is the way the distance is calculated.
3. FCM uses Euclidean distances
4. GK uses Mahalanobis distances

C-Means Problem

1. C-Means and K-Means produce spherical clusters.



Deforming Space - I

1. An arbitrary, positive definite and symmetric matrix $A \in \mathcal{R}^{p \times p}$ induces a scalar product $< x, y >= x^T Ay$
2. For instance the matrix $A = \left[\begin{array}{cc} 1/4 & 0 \\ 0 & 4 \end{array} \right]$ provides

$$\left\| \begin{array}{c} x \\ y \end{array} \right\|_A^2 = (x\ y) \left[\begin{array}{cc} 1/4 & 0 \\ 0 & 4 \end{array} \right] \left(\begin{array}{c} x \\ y \end{array} \right) = \left(\begin{array}{c} x^2/4 \\ 4y^2 \end{array} \right) = \left(\begin{array}{c} (x/2)^2 \\ (2y)^2 \end{array} \right)$$

3. This corresponds to a deformation of the unit circle to an ellipse with a double diameter in the x-direction and a half diameter in the y-direction.

Deforming Space - II

1. If there is a priori knowledge about the data elliptic clusters can be recognized.
2. If each has its own matrix the different elliptic clusters can be obtained.

GK details

1. Mahalanobis distance is calculated as

$$d_{ik}^2 = (\mathbf{x}_i - \mathbf{v}_k)^T A_i (\mathbf{x}_i - \mathbf{v}_k)$$

2. It is possible to prove that the matrices A_i are given by

$$A_i = \sqrt[p]{\det S_i} S_i^{-1}$$

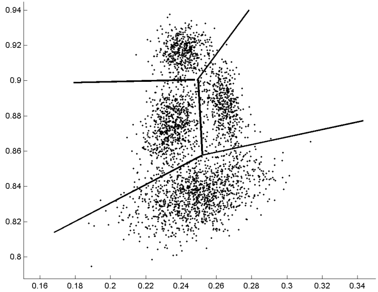
3. S_i is the fuzzy covariance matrix and equal to

$$S_i = \sum_{e=1}^{n} \mu_{ie}^m (\mathbf{x_e} - \mathbf{v_i})(\mathbf{x_e} - \mathbf{v_i})^T$$

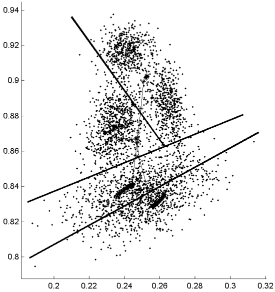
GK Comments

1. In addition to cluster centres each cluster is characterized by a symmetric and positive definite matrix A.
2. The clusters are hyper-ellipsoids on the $\mathbb{R}^l$.
3. The hyper-ellipsoids have approximately the same size.
4. In order to be possible to calculate S^{-1} the number of samples n must be at least equal to the number of dimensions l plus 1.

GK Results



0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
0.16 0.18 0.2 0.22 0.24 0.26 0.28 0.3 0.32 0.34



0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
0.2 0.22 0.24 0.26 0.28 0.3 0.32

Section Summary

Introduction

1. It is also known as Gaussian Mixture Decomposition.
2. It is similar to the FCM method
3. The Gauss distance is used instead of Euclidean distance.
4. The clusters no longer have a definite shape and may have various sizes.

Gauss Distance

1. Gauss distance

$$d_{ie} = \frac{1}{P_i}\sqrt{\det(A_i)}\ \exp\left(\frac{1}{2}(\mathbf{x_e} - \mathbf{v_i})^T A^{-1}(\mathbf{x_e} - \mathbf{v_i})\right)$$

2. Cluster centre

$$\mathbf{v}_i = \frac{\displaystyle\sum_{e=1}^{n} \mu_{ie} \cdot \mathbf{x}_e}{\displaystyle\sum_{e=1}^{n} \mu_{ie}}$$

Gauss Distance

1. $$A_i = \frac{\sum_{e=1}^{n} \mu_{ie}^{m} (\mathbf{x_e} - \mathbf{v_i})(\mathbf{x_e} - \mathbf{v_i})^T}{\sum_{e=1}^{n} \mu_{ie}^{m}}$$

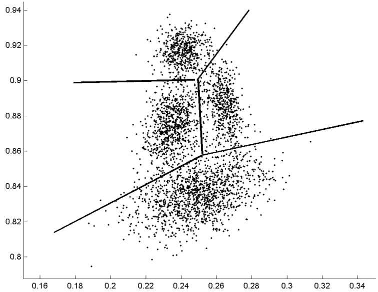
2. Probability that an element $\mathbf{x_e}$ belongs to the cluster i

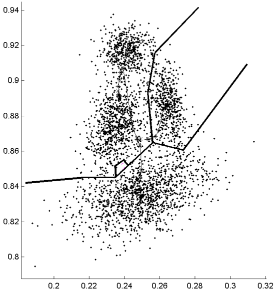
$$P_i = \frac{\sum_{e=1}^{n} \mu_{ie}^{m}}{\sum_{e=1}^{n} \sum_{i=1}^{c} \mu_{ie}^{m}}$$

GG Comments

1. P_i is a parameter that influences the size of a cluster.
2. Bigger clusters attract more elements.
3. The exponential term makes more difficult to avoid local minima.
4. Usually another clustering method is used to initialise the partition matrix U.

GG Results


0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
0.16 0.18 0.2 0.22 0.24 0.26 0.28 0.3 0.32 0.34


0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
0.2 0.22 0.24 0.26 0.28 0.3 0.32

Section Summary

Source

1. Algorithm presented in: **Finding groups in Data: An Introduction to clusters analysis**, L. Kaufman and P. J. Rousseeuw, John Wiley & Sons

Methods

1. K-means is sensitive to outliers since an object with an extremely large value may distort the distribution of data.
2. Instead of taking the mean value the most centrally object (medoid) is used as reference point.
3. The algorithm minimizes the sum of dissimilarities between each object and the medoid (similar to k-means)

Strategies

1. Find k-medoids arbitrarily.
2. Each remaining object is clustered with the medoid to which is the most similar.
3. Then iteratively replaces one of the medoids by a non-medoid as long as the quality of the clustering is improved.
4. The quality is measured using a cost function that measures the sum of the dissimilarities between the objects and the medoid of their cluster.

Reassignment

1. Each time a reassignment occurs a difference in square-error J is contributed.
2. The cost function J calculates the total cost of replacing a current medoid by a non-medoid.
3. If the total cost is negative then m_j is replaced by m_{random}, otherwise the replacement is not accepted.

Phases

1. Build phase: an initial clustering is obtained by the successive selection of representative objects until c (number of clusters) objects have been found.
2. Swap phase: an attempt to improve the set of the c representative objects is made.

Build Phase

1. The first object is the one for which the sum of dissimilarities to all objects is as small as possible.
2. This is most centrally located object.
3. At each subsequent step another object that decreases the objective function is selected.

Build Phase next steps I

1. Consider an object e_i (candidate) which has not yet been selected.
2. Consider a non selected object e_j.
3. Calculate the difference C_{ji}, between its dissimilarity $D_j = d(e_n, e_j)$ with the most similar previously selected object e_n, and its dissimilarity $d(e_i, e_j)$ with object e_i.
4. If $C_{ij} = D_j - d(e_i, e_j)$ is positive then object e_j will contribute to select object e_i, $C_{ij} = \max(D_j - d(e_i, e_j), 0)$
5. If $C_{ij} > 0$ then e_j is closer to e_i than any other previously selected object.

Build Phase next steps II

1. Calculate the total gain G_i obtained by selecting object e_i
   $$G_i = \sum_j (C_{ji})$$
2. Choose the not yet selected object e_i which maximizes $G_i = \sum_j (C_{ji})$
3. The process continues until c objects have been found.

Swap Phase

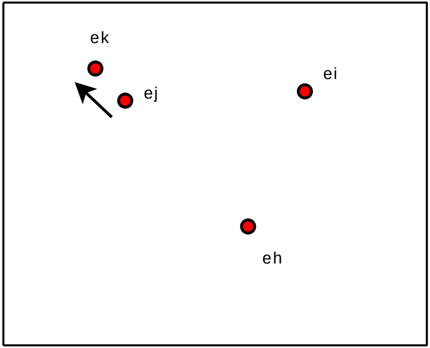
1. It is attempted to improve the set of representative elements.
2. Consider all pairs of elements (i, h) for which e_i has been selected and e_h has not.
3. What is the effect of swapping e_i and e_h?
4. Consider the objective function as the sum of dissimilarities between each element and the most similar representative object.

Swap Phase - possibility a

1. What is the effect of swapping e_i and e_h?
2. Consider a non selected object e_j and calculate its contribution C_{jih} to the swap:
3. If e_j is more distant from both e_i and e_h than from one of the other representatives, e.g. e_k, so $C_{ijh} = 0$
4. So e_j belongs to object e_k, sometimes referred as the medoid m_k and the swap will not change the quality of the clustering.
5. Remember: positive contributions decrease the quality of the clustering.

Swap Phase - possibility a

1. Object e_j belongs to medoid $e_k (i \neq k)$.
2. If e_i is replaced by e_h and e_j is still closer to e_k, then $C_{jih} = 0$.

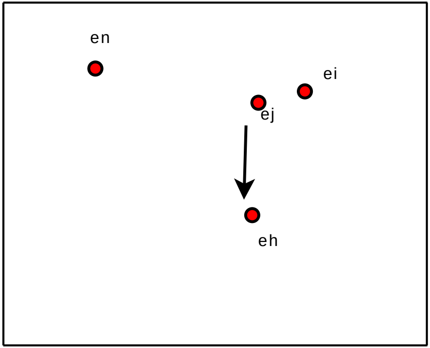


Swap Phase - possibility b

1. If e_j is not further from e_i than from any one of the other representative $(d(e_i, e_j) = D_j)$, two situations must be considered:
2. e_j is closer to e_h than to the second closest representative e_n, $d(e_j, e_h) < d(e_j, e_n)$ then $C_{jih} = d(e_j, e_h) - d(e_j, e_i)$.
3. Contribution C_{jih} can either positive or negative.
4. If element e_j is closer to e_i than to e_h the contribution is positive, the swap is not favourable.

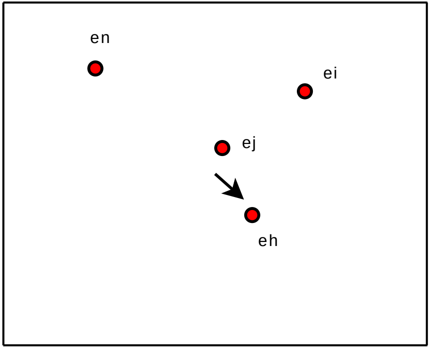
Swap Phase - possibility b.1+

1. Object e_j belongs to medoid e_i.
2. If e_i is replaced by e_h and e_j is close to e_i than e_h the contribution is positive, $Cjih > 0$.



Swap Phase - possibility b.1-

1. Object e_j belongs to medoid e_i.
2. If e_i is replaced by e_h and e_j is not closer to e_i than e_h the contribution is negative. $C_{jih<0}$.

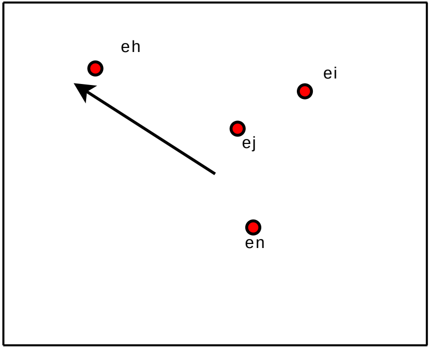


Swap Phase - possibility b2

1. e_j is at least as distant from e_h than from the second closest representative $d(e_j, e_h) \geq d(e_j, e_n)$ then $C_{jih} = d(e_j, e_n) - d(e_j, e_i)$
2. The contribution is always positive because it is not advantageous to replace e_i by an e_h further away from e_j than from the second best closest representative object.

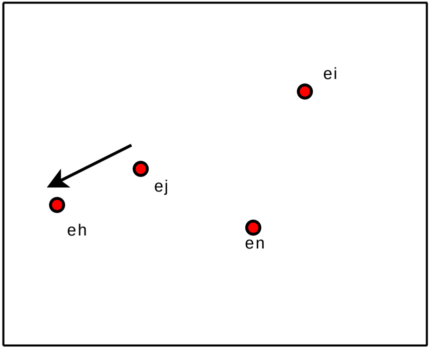
Swap Phase - possibility b.2

1. Object e_j belongs to medoid e_i.
2. If e_i is replaced by e_h and e_j is further from e_h than e_n, the contribution is always positive. $C_{jih>0}$.



Swap Phase - possibility c

1. e_j is more distant from e_i than from at least one of the other representative objects (e_n) but closer to e_h than to any representative object, then $C_{jih} = d(e_j, e_h) - d(e_j, e_i)$



Comparisons

1. K-medoids is more robust than k-means in presence of noise and outliers.
2. K-means is less costly in terms of processing time.

The End