K-Medoids algorithm

-Select *k* of the *n* data point as the medoids
-Associate each data point to the closest medoid
-compute the sum of all associated data point -> cost
-Swap medoid (m) and non-medoid (o)
|
V
loop
-Associate each data point to the closest swaped medoid
-compute the sum of all associated data point -> newcost
if newcost > cost , the algorithm terminates

By M.Aek Progs

https://www.facebook.com/M.Aek.Progs.Angedevil.AD/

https://www.youtube.com/channel/UC6AhJIORlsp56XDwqfNSTsg

https://github.com/Angedevil-AD

Compute Cost

$$c = \sum_{j=0}^{n} \arg\min_i \| o_j - m_i \|^2$$

Where:
c: cost
n: size of data point
o: data point
m: medoid

Convergence

$$Convergence = \begin{cases} 1, \ if\ C^{t} > C^{t-1} \\ 0, \ Otherwise \end{cases}$$

Exemple

dataset o: 30,19,39,18,10,14 n=6

Generate k medoids from dataset randomely:

lets asume k =2 --> 2 cluster & 2 medoids

medoids m: 19,18
Cluster(0) = 19
Cluster(1) = 18

-Associate each data point to the closest medoid :
using euclidean distance:
note!!! medoids in the dataset bypassed

$(30-19)^2 = 121$
$(30-18)^2 = 144$

$(39-19)^2 = 400$
$(39-18)^2 = 441$

$(10-19)^2 = 81$
$(10-18)^2 = 64$

$(14-19)^2 = 25$
$(14-18)^2 = 16$

parse min distance:
121,400,64,16

Cluster(0) = 19,30,39
Cluster(1) = 18,64,16

Compute Cost:

c(t)= 121+400+64+16 =601

Swap medoids:
Generate k medoids from dataset randomely:

medoids m: 10,19
Cluster(0) = 10
Cluster(1) = 19

-Associate each data point to the closest medoid :

$(30-10)^2 = 400$
$(30-19)^2 = 121$

$(39-10)^2 = 841$
$(39-19)^2 = 400$

$(18-10)^2 = 64$
$(18-19)^2 = 1$

$(14-10)^2 = 16$
$(14-19)^2 = 25$

parse min distance:
121,400,1,16

Cluster(0) = 10,16
Cluster(1) = 19,30,39,18

Compute Cost:

c(t1)= 121+400+1+16 =538

c(t1) < c(t) , Not Converged

copy c(t1) -> c(t)

continue.......

Swap medoids:
Generate k medoids from dataset randomely:

medoids m: 39,10
Cluster(0) = 39
Cluster(1) = 10

-Associate each data point to the closest medoid :

$(30-39)^2 = 81$
$(30-10)^2 = 400$

$(19-39)^2 = 400$
$(19-10)^2 = 81$

$(18-39)^2 = 441$
$(18-10)^2 = 64$

$(14-39)^2 = 625$
$(14-10)^2 = 16$

parse min distance:
81,81,64,16

Cluster(0) = 39,30
Cluster(1) = 10,19,18,14

Compute Cost:

c(t1)= 81+81+64+16 =242

c(t1) < c(t) , Not Converged

copy c(t1) -> c(t)

Swap medoids:
Generate k medoids from dataset randomely:

medoids m: 30,18
Cluster(0) = 30
Cluster(1) = 18

-Associate each data point to the closest medoid :

$(19-30)^2 = 121$
$(19-18)^2 = 1$

$(39-30)^2 = 81$
$(39-18)^2 = 441$

$(10-30)^2 = 400$
$(10-18)^2 = 64$

$(14-30)^2 = 256$
$(14-18)^2 = 16$

parse min distance:
1,81,64,16

Cluster(0) = 30,81
Cluster(1) = 18,19,10,14

Compute Cost:

c(t1)= 81+1+64+16 =162

c(t1) < c(t) , Not Converged

copy c(t1) -> c(t)

Swap medoids:
Generate k medoids from dataset randomely:

medoids m: 10,30
Cluster(0) = 10
Cluster(1) = 30

-Associate each data point to the closest medoid :

$(19-10)^2 = 81$
$(19-30)^2 = 121$

$(39-10)^2 = 841$
$(39-30)^2 = 81$

$(18-10)^2 = 64$
$(18-30)^2 = 144$

$(14-10)^2 = 16$
$(14-30)^2 = 256$

Time Complexity:

$$O(n^2 k^2)$$

parse min distance:
81,81,64,16

Cluster(0) = 10,19,18,14
Cluster(1) = 30,39

Compute Cost:

c(t1)= 81+81+64+16 =242

c(t1) > c(t) , Converged

medoids m = 10,30

Cluster(0) = 10,19,18,14
Cluster(1) = 30,39

--Algorithm Terminate--

Cluster 0	**10 , 19 , 18 , 14**
Cluster 1	**30 , 39**

K medoids is more better than k means for handling outliers