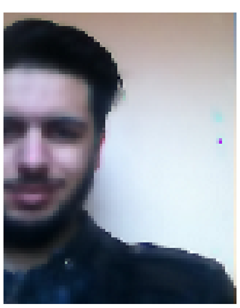
K-means algorithm

overview:

k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. k-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, better Euclidean solutions can be found using k-medians and k-medoids.

The problem is computationally difficult (NP-hard); however, efficient heuristic algorithms converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both k-means and Gaussian mixture modeling. They both use cluster centers to model the data; however, k-means clustering tends to find clusters of comparable spatial extent, while the Gaussian mixture model allows clusters to have different shapes.



delphi & c++ programmation
For more information:

contact us at: fb ..gitH ..or .. yt

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

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

https://github.com/Angedevil-AD

Assignement Step:

xp one set of observation
xp belong to (x0,x1,x2.....xn)

m :set of K means where m belong to (m0,m1,m2...mK)
K: number of means

size of S = Size of Means = K

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \quad \forall\, j,\, 1 \leq j \leq k \right\}$$

assign each (xp) to (Si) with the nearest mean:
calc the nearest mean using the euclidean distance

Update Step:

mi: set of K means
t+1: iteration
x set of observation assigned to S

$$m_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{xj \in S_i^{(t)}} x_j$$

the algorithm has converged when the assignement no longer change

$$\boxed{\begin{array}{l} S_i^{(t)} = S_i^{(t\text{-}1)} \text{ , clusters not changed.. Converged} \\ S_i^{(t)} \neq S_j^{(t\text{-}1)} \text{ , cluster changed ... Not Converged} \end{array}}$$

Now lets simplify this algorithm:

Solve the K means EQs :

set of observation
length = 6
$(x0,x1,x2,x3,x4,x5) = (4,11,40,10,8,20)$

set of means
length = K = 3
$K <= 6$
$(m0,m1,m2)$
generate 3 means randomy from set of observation:
ex: means (11,8,40)

assigment step: iteration (t) = 0
clusters length = K
S0 = current cluster
S1= previous cluster

for each set of observation

$xp = x0 = 4$
$i = 0 \quad j = 1 \qquad 1 <= j < k$ --> from 1 to 2
$i = (xp - m(i))^2 < (xp - m(j))^2 \ ? \ i:j$
$i = (4 - 11)^2 < (4-8)^2 \ ? \ i:j$
$i = 7^2 < 4^2 \ ? \ i: i+1$ .... 7^2 not $< 4^2$ so $i = j$
$i = 1$

$i=1 \ j = 2$
$i = (xp - m(i))^2 < (xp - m(j))^2 \ ? \ i:j$
$i = (4 - 8)^2 < (4-40)^2 \ ? \ i:j$
$i = 4^2 < 36^2 ? \ i: j$ .... $4^2 < 36^2$ so $i = i$
$i=1$
assign xp to S(i)
S0(i) = xp ... S0(1) = 4

$xp = x1 = 11$
$i = 0 \quad j = 1 \qquad 1 <= j < k$ --> from 1 to 2
$i = (xp - m(i))^2 < (xp - m(j))^2 \ ? \ i:j$
$i = (11- 11)^2 < (11-8)^2 \ ? \ i:j$
$i = 0 < 3^2 \ ? \ i: j$ .... $0 < 9$ so $i = i$
$i = 0$

$i=0 \ j = 2$
$i = (xp - m(i))^2 < (xp - m(j))^2 \ ? \ i:j$
$i = (11 - 11)^2 < (11-40)^2 \ ? \ i:j$
$i = 9 < 29^2 \ ? \ i: j$ .... $9 < 29^2$ so $i = i$
$i=0$
assign xp to S(i)
S0(i) = xp ... S0(0) = 11

$xp = x2 = 40$
$i = 0 \quad j = 1 \qquad 1 <= j < k$ --> from 1 to 2
$i = (xp - m(i))^2 < (xp - m(j))^2 \ ? \ i:j$
$i = (40- 11)^2 < (40-8)^2 \ ? \ i:j$
$i = 29^2 < 32^2 \ ? \ i: j$ .... $29^2 < 32^2$ so $i = i$
$i = 0$

$i=0 \ j = 2$
$i = (xp - m(i))^2 < (xp - m(j))^2 \ ? \ i:j$
$i = (40 - 11)^2 < (40-40)^2 \ ? \ i:j$
$i = 29^2 < 0 \ ? \ i: j$ ..29^2 not < 0 so $i = j$
$i=2$
assign xp to S(i)
S0(i) = xp ... S0(2) = 40

$xp = x3 = 10$
$i = 0 \quad j = 1 \qquad 1 <= j < k$ --> from 1 to 2
$i = (xp - m(i))^2 < (xp - m(j))^2 \ ? \ i:j$
$i = (10- 11)^2 < (10-8)^2 \ ? \ i:j$
$i = 1 < 2^2 \ ? \ i: j$ .... $0 < 2^2$ so $i = i$
$i = 0$

$i=0 \ j = 2$
$i = (xp - m(i))^2 < (xp - m(j))^2 \ ? \ i:j$
$i = (10- 11)^2 < (10-40)^2 \ ? \ i:j$
$i = 1 < 30^2 \ ? \ i: j$ .... $4< 100$ so $i = i$
$i=0$
assign xp to S(i)
S0(i) = xp ... S0(0) = 11,10

xp = x4= 8
i = 0 j = 1   1 <=j <k --> from 1 to 2
i = (xp - m(i))^2 < (xp - m(j))^2 ? i:j
i = (8- 11)^2 < (8-8)^2 ? i:j
i = 3^2 < 0 ? i: j .... 3 not < 0 so i = j
i = 1

i=1 j = 2
i = (xp - m(i))^2 < (xp - m(j))^2 ? i:j
i = (8- 8)^2 < (8-40)^2 ? i:j
i = 0 < 32^2? i: j.... 0< 32^2 so i = i
i=1
assign xp to S(i)
S0(i) = xp ... S0(1) = 4,8

xp = x5= 20
i = 0 j = 1   1 <=j <k --> from 1 to 2
i = (xp - m(i))^2 < (xp - m(j))^2 ? i:j
i = (20- 11)^2 < (20-8)^2 ? i:j
i = 9^2 < 12^2? i: j .... 9^2< 12^2 so i = i
i = 0

i=0 j = 2
i = (xp - m(i))^2 < (xp - m(j))^2 ? i:j
i = (20- 11)^2 < (20-40)^2 ? i:j
i = 9^2< 20^2? i: j .... 9^2 < 20^2 so i = j
i=0
assign xp to S(i)
S0(i) = xp ... S0(0) = 11,10,20

we get :
S0(0) = 11,10,20
S0(1) = 4,8
S0(2) = 40

S1(0)=null S1(1)=null S1(2) = null ..S0 is not equal to S1 .continue until S1 = S0

Copy S0 to S1 ... S1(0) = 11,10,20 ..S1(1) = 4,8 .. S1(2) = 40
update step:
mi= S0(j)+ S0(j+1).....+S0(n) / n

m0 = 11+10+20 / 3 = 13,6666
m1 = 4 +8 / 2 = 6
m2 = 40/1=40

pass to t+1
(x0,x1,x2,x3,x4,x5) = (4,11,40,10,8,20)

means is updated
means (13.6666,6,40)

for each set of observation
xp = x0 = 4
i = 0 j = 1   1 <=j <k --> from 1 to 2
i = (4 - 13.6666,)^2 < (4-6)^2 ? i:j
i = 9,6666^2 < 2^2 ? i: j .... not < ...so i =j
i = 1

i=1 j = 2
i = (xp - m(i))^2 < (xp - m(j))^2 ? i:i+1
i = (4 - 6)^2 < (4-40)^2 ? i:i+1
i = 2^2 < 36^2 ? i: j .... i = i
i=1
assign xp to S(i)
S0(i) = xp ... S0(1) = 4

xp = x1= 11
i = 0 j = 1   1 <=j <k --> from 1 to 2
i = 2,6666^2 < 5^2 ? i: j ...so i =i
i = 0

i=0 j = 2
1
i = 2,6666^{22} < 29^2 ? i: j .... i = i
i=0
assign xp to S(i)
S0(i) = xp ... S0(0) = 11

xp = x2= 40
i = 0 j = 1   1 <=j <k --> from 1 to 2
i = $26{,}3333333333333^2 < 34^2$ *? i: j   ...so i =i*
i = 0

i=0 j = 2
i = $26{,}3333333333333^2 < 0^2$ *? i: j ..not <.. so i = j*
i=2
assign xp to S(i)
S0(i) = xp ... S0(2) = 40

xp = x3= 10
i = 0 j = 1   1 <=j <k --> from 1 to 2
i = $3{,}66666666666667^2 < 4^2$ *? i: j   ...so i =i*
i = 0

i=0 j = 2
i = $3{,}66666666666667 < 30^2$ *? i: j so i = i*
i=0
assign xp to S(i)
S0(i) = xp ... S0(0) = 11,10

xp = x4= 8

i = 0 j = 1   1 <=j <k --> from 1 to 2
i = $5{,}66666^2 < 2^2$ *? i: j   ...not < ..so i =j*
i = 1

i=1 j = 2
i = $2^2 < 32^2$ *? i: j so i = i*
i=1
assign xp to S(i)
S0(i) = xp ... S0(1) = 4,8

xp = x4= 20

i = 0 j = 1   1 <=j <k --> from 1 to 2
i = $6{,}33333^2 < 14^2$ *? i: j   ... ..so i =i*
i = 1

i=0 j = 2
i = $6{,}33333^2 < 20^2$ *? i: j so i = i*
i=0
assign xp to S(i)
S0(i) = xp ... S0(0) = 11,10,20

we get :
S0(0) = 11,10,20
S0(1) = 4,8
S0(2) = 40

check previous cluster S1
S1(0) = 11,10,20 ..S1(1) = 4,8 .. S1(2) = 40

S0 = S1 .... Convergence = ok

Centroid = means
m0 = 13,66666
m1 = 6
m2 = 40

time Complexity:

$$O(N^{dk+1})$$