# (AP)Affinity Propagation algorithm

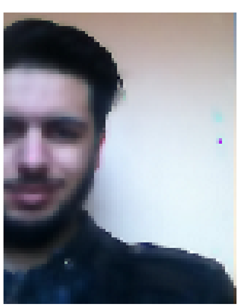
**Overview:**

**In statistics and data mining, affinity propagation (AP) is a clustering algorithm based on the concept of "message passing" between data points. Unlike clustering algorithms such as k-means or k-medoids, affinity propagation does not require the number of clusters to be determined or estimated before running the algorithm. Similar to k-medoids, affinity propagation finds "exemplars," members of the input set that are representative of clusters**

---

**Algo:**

**1-compute similarities**
**2-compute responsibilities**
**3-compute availabilities**




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

**x: set of data points**
**N: size of dataset .. where ...0<= i <N & 0 <=k <N**
**availability , similarity, responsibility matrix = NxN**

**1- Similarity Equation:**

$$s(i,k) = -\|x_i - x_k\|^2$$



**Diagonal s(i,i) ,initialized to the median similarity**

---

**2-Responsibility Equation:**
**responsibilities matrix initialized to 0**

$$r(i,k) \leftarrow s(i,k) - \max_{k \neq k'}\{a(i,k') + s(i,k')\}$$

**3-Availabilities:**

$$a(i,k) \leftarrow \min\left(0\,,\ r(k,k) + \sum_{i' \notin (i,k)} \max(0, r(i',k))\right) \ for\ i \neq k$$

$a(k,k)$ *diagonal:*

$$a(k,k) \leftarrow \sum_{i' \neq k} \max(\,0\,,\ r(i'\,,\,k))$$



**finally ,**
**algorithm terminate if clusters unchanged after infinite number of iteration (use break to finaliz the iteration loop) , or after predetermined number of iteration is reached**

**exemplar extracted from diagonal sum of r & a , where r(i,i) + a(i,i) > 0**

**lets explain the 3 equation with exemple:**

dataset =
one dimension
0.90 ,0.15 ,0.62 ,0.8 ,0.27 ,0.18 ,0.30
len ...N= 7

compute similarities:
as we know
$s(i,k) = - || xi - xk||^2$

$0<= i <N$
$0<=k <N$

similarities matrix s(i,k)

$$\begin{bmatrix}
0.00 , & -0.56 , & -0.08 , & -0.01 , & -0.40 , & -0.52 , & -0.36 , \\
-0.56 , & 0.00 , & -0.22 , & -0.42 , & -0.01 , & -0.00 , & -0.02 , \\
-0.08 , & -0.22 , & 0.00 , & -0.03 , & -0.12 , & -0.19 , & -0.10 , \\
-0.01 , & -0.42 , & -0.03 , & 0.00 , & -0.28 , & -0.38 , & -0.25 , \\
-0.40 , & -0.01 , & -0.12 , & -0.28 , & 0.00 , & -0.01 , & -0.00 , \\
-0.52 , & -0.00 , & -0.19 , & -0.38 , & -0.01 , & 0.00 , & -0.01 , \\
-0.36 , & -0.02 , & -0.10 , & -0.25 , & -0.00 , & -0.01 , & 0.00
\end{bmatrix}$$

$s(0,0) = -(0.90-0.90)^2 = 0.0$
$s(0,1) = -(0.90-0.15)^2 = -0.56$
$s(0,2) = -(0.90-0.62)^2 = -0.08$
.
.
$s(6,6) = -(0.30-0.30)^2 =0.0$
put result in Matrix NxN

fill Diagonal(s(i,i) with median value of similarity matrix:

0.00 , -0.56 , -0.08 , -0.01 , -0.40 , -0.52 , -0.36 ,
-0.56 , 0.00 , -0.22 , -0.42 , -0.01 , -0.00 , -0.02 ,
-0.08 , -0.22 , 0.00 , -0.03 , -0.12 , -0.19 , -0.10 ,
-0.01 , -0.42 , -0.03 , 0.00 , -0.28 , -0.38 , -0.25 ,
-0.40 , -0.01 , -0.12 , -0.28 , 0.00 , -0.01 , -0.00 ,
-0.52 , -0.00 , -0.19 , -0.38 , -0.01 , 0.00 , -0.01 ,
-0.36 , -0.02 , -0.10 , -0.25 , -0.00 , -0.01 , 0.00

before calc median value we must sort the matrix :

0,00 0,00 0,00 0,00 0,00 0,00 0,00

0,00 0,00 0,00 0,00 -0,01 -0,01 -0,01

-0,01 -0,01 -0,01 -0,01 -0,01 -0,02 -0,02

-0,03 -0,03 -0,08 $\boxed{-0,08}$ -0,10 -0,10 -0,12

-0,12 -0,19 -0,19 -0,22 -0,22 -0,25 -0,25

-0,28 -0,28 -0,36 -0,36 -0,38 -0,38 -0,40

-0,40 -0,42 -0,42 -0,52 -0,52 -0,56 -0,56

median = sorted_similarities(N/2,N/2)
= -0.08

final similarities matrix :

$$
\begin{bmatrix}
-0.08\ , & -0.56\ , & -0.08\ , & -0.01\ , & -0.40\ , & -0.52\ , & -0.36\ , \\
-0.56\ , & -0.08\ , & -0.22\ , & -0.42\ , & -0.01\ , & -0.00\ , & -0.02\ , \\
-0.08\ , & -0.22\ , & -0.08\ , & -0.03\ , & -0.12\ , & -0.19\ , & -0.10\ , \\
-0.01\ , & -0.42\ , & -0.03\ , & -0.08\ , & -0.28\ , & -0.38\ , & -0.25\ , \\
-0.40\ , & -0.01\ , & -0.12\ , & -0.28\ , & -0.08\ , & -0.01\ , & -0.00\ , \\
-0.52\ , & -0.00\ , & -0.19\ , & -0.38\ , & -0.01\ , & -0.08\ , & -0.01\ , \\
-0.36\ , & -0.02\ , & -0.10\ , & -0.25\ , & -0.00\ , & -0.01\ , & -0.08
\end{bmatrix}
$$

- we done with similarities matrix.
lets move to responsibilities and availabilities

these two matrix , must updated continuouly as long as the algorithm not terminated, therfore we have to determine the number of iteration , or use an infinite loop and break if clusters, unchanged

for i=0 to infinite
responsibilities(...)
availabilities(...)
if curent clusters = previous clusters ..break

## -responsibility:

r(i,k) = s(i,k) - max(s(i,k')+a(i,k')) wherre k not equal k'

compute max value in s(i,k')+a(i,k')

0<= i <N, 0<=k <N , 0<= k' <N
for i = 0 , k = 0, k'= 0
k = k' ..we know that k must not equal to k' ..so this step will bypassed

move to

---------------------------------
for i = 0 , k = 0, k'= 1
k<>k' ... fine
s(i,k') = -0.56 ...v(alue from similarities matrix)
a(i,k') = 0 ... availability is not yet filled ... is initialized to 0 for first use
s+k=-0.56
---------------------------------

for i = 0 , k = 0, k'= 2
k<>k' ... fine
s(i,k') = -0.08
a(i,k') = 0
s+k=-0.08
---------------------------------

for i = 0 , k = 0, k'= 3
k<>k' ... fine
s(i,k') = -0.01
a(i,k') = 0
s+k=-0.01
---------------------------------

for i = 0  , k = 0, k’= 4
k<>k’  ... fine
s(i,k’) = -0.40
a(i,k’) = 0
s+k=-0.40
---------------------------------
for i = 0  , k = 0, k’= 5
k<>k’  ... fine
s(i,k’) = -0.52
a(i,k’) = 0
s+k=-0.52
---------------------------------
for i = 0  , k = 0, k’= 6
k<>k’  ... fine
s(i,k’) = -0.36
a(i,k’) = 0
s+k=-0.36
---------------------------------

max value in : -0.56 ,  -0.08 ,  -0.01 ,  -0.40 ,  -0.52 ,  -0.36  is -0.01
maxval = -0.01

r(i,k) = s(i,k) - maxval = -0.08 - (-0.01) = -0.07

continue:

for i = 0  , k = 1, k’= 0
k<>k’  ... fine
s(i,k’) = -0.08
a(i,k’) = 0
s+k=-0.08

---------------------------------
for i = 0  , k = 1, k’= 1
k=k’  ... bypass
---------------------------------
for i = 0  , k = 1, k’= 2
k<>k’  ... fine
s(i,k’) = -0.08
a(i,k’) = 0
s+k=-0.08

--------------------------------
for i = 0 , k = 1, k’= 3 k<>k’ ... fine
s(i,k’) = -0.01
a(i,k’) = 0
s+k=-0.01

.
.continue until
--------------------------------
for i = 0 , k = 1, k’= 6 k<>k’ ... fine
s(i,k’) = -0.36
a(i,k’) = 0
s+k=-0.36

max value in : -0.08, -0.08 , -0.01 , -0.40 , -0.52 , -0.36 is -0.01 maxval = -0.01

r(i,k) = s(i,k) - maxval = -0.56 - (-0.01) = -0.55

continue until
for i = 6 , k = 6, k’= 6

put all result in matrix, and before do that , we need apply dump factor on result, where dump factor [0...1]
ex: dump factor = 0.3
new matrix = (1-dump factor) * current matrix + dump factor * previous matrix for
matrix[0][0] = (1-0.3)*0.07 - 0.3*0 = -0.05

responsibilities matrix:

$$
\begin{bmatrix}
-0.05, & -0.39, & -0.05, & 0.05, & -0.27, & -0.36, & -0.25, \\
-0.39, & -0.05, & -0.15, & -0.30, & -0.01, & 0.01, & -0.02, \\
-0.03, & -0.13, & -0.03, & 0.03, & -0.06, & -0.11, & -0.05, \\
0.02, & -0.29, & -0.02, & -0.05, & -0.19, & -0.26, & -0.17, \\
-0.28, & -0.01, & -0.09, & -0.20, & -0.05, & -0.01, & 0.01, \\
-0.36, & 0.01, & -0.13, & -0.27, & -0.01, & -0.05, & -0.01, \\
-0.25, & -0.02, & -0.07, & -0.17, & 0.01, & -0.01, & -0.05
\end{bmatrix}
$$

now compute availabilities:

**-availabilities:**

a(i,k) = min(0, r(k,k) + sum(  max(0,r(i’,k))  ) i not equal to k and i’ not belong to (i,k)

compute max value in 0,r(i’,k)
0<= i <N,   0<=k <N ,   0<= i’ <N
for i = 0  , k = 0, k’= 0
i = i’ bypass
---------------------------------
for i = 0  , k = 0, i’= 1
i<>i’  ... fine
r(i’,k) = -0.39...(value from responsibilities matrix)
max(0,-0.39) = 0
---------------------------------
for i = 0  , k = 0, i’= 2 i<>i’
... fine
r(i’,k) = -0.03
. max(0,-0.03) = 0
---------------------------------
for i = 0  , k = 0, i’= 3 i<>i’
... fine
r(i’,k) = 0.02
max(0,0.02) = 0.02
---------------------------------
for i = 0  , k = 0, i’= 4
i<>i’  ... fine
r(i’,k) = -0.28
max(0,-0.28) = 0
---------------------------------
for i = 0  , k = 0, i’= 5
i<>i’  ... fine
r(i’,k) = -0.36
max(0,-0.36) = 0
---------------------------------
for i = 0  , k = 0, i’= 6
i<>i’  ... fine
r(i’,k) = -0.25

max(0,-0.25) = 0

sum all result : 0+0+0.02+0+0+0   =   0.02
add sum to r(k,k) ..r(k,k) = r(0,0) = -0.05
-0.05 + 0.02= -0.03
a(i,k) = min(0,-0.03) = -0.03 <----------

now set diagonal a(k,k) a(k,k) = sum( max(0,r(i’,k))) sum( max(0,r(i’,k))) = 0.02
a(k,k) = 0.02

dont forgot to apply dump factor to availabilities matrix ........
continue until
for i = 6 , k = 6, i’= 6

you get as result :

availabilities matrix

$$
\begin{bmatrix}
0.01 , & -0.03 , & -0.02 , & -0.01 , & -0.03 , & -0.03 , & -0.03 , \\
-0.02 , & 0.00 , & -0.02 , & 0.00 , & -0.03 , & -0.04 , & -0.03 , \\
-0.02 , & -0.03 , & 0.00 , & 0.00 , & -0.03 , & -0.03 , & -0.03 , \\
-0.03 , & -0.03 , & -0.02 , & 0.06 , & -0.03 , & -0.03 , & -0.03 , \\
-0.02 , & -0.03 , & -0.02 , & 0.00 , & 0.01 , & -0.03 , & -0.04 , \\
-0.02 , & -0.04 , & -0.02 , & 0.00 , & -0.03 , & 0.01 , & -0.03 , \\
-0.02 , & -0.03 , & -0.02 , & 0.00 , & -0.04 , & -0.03 , & 0.00
\end{bmatrix}
$$

we done with the first iteration .....
calc availabilities + responsibilities & extract exemplar from diagonal. availabilities + responsibilities matrix:

-0.04 , -0.42 , -0.07 , 0.04 , -0.30 , -0.39 , -0.28 ,
-0.42 , -0.05 , -0.18 , -0.30 , -0.04 , -0.03 , -0.05 ,
-0.05 , -0.17 , -0.03 , 0.03 , -0.09 , -0.14 , -0.08 ,
-0.02 , -0.32 , -0.04 , 0.01 , -0.22 , -0.29 , -0.20 ,
-0.30 , -0.04 , -0.11 , -0.20 , -0.05 , -0.04 , -0.03 ,
-0.38 , -0.03 , -0.16 , -0.27 , -0.04 , -0.05 , -0.04 ,
-0.27 , -0.05 , -0.09 , -0.17 , -0.03 , -0.04 , -0.05 ,

check if any elemnent of the diagonal = positive ?

-0.04 , -0.42 , -0.07 , 0.04 , -0.30 , -0.39 , -0.28 ,
-0.42 , -0.05 , -0.18 , -0.30 , -0.04 , -0.03 , -0.05 ,
-0.05 , -0.17 , -0.03 , 0.03 , -0.09 , -0.14 , -0.08 ,
-0.02 , -0.32 , -0.04 , 0.01 , -0.22 , -0.29 , -0.20 ,
-0.30 , -0.04 , -0.11 , -0.20 , -0.05 , -0.04 , -0.03 ,
-0.38 , -0.03 , -0.16 , -0.27 , -0.04 , -0.05 , -0.04 ,
-0.27 , -0.05 , -0.09 , -0.17 , -0.03 , -0.04 , -0.05 ,

0.01 = positive

(3,3) = 0.01

exemplar = 0.01

keep updating availabilities & responsibilities by passing to iteration 2,3,4....until n

check result in iteration 2:

-0.06 , -0.68 , -0.12 , 0.07 , -0.46 , -0.61 , -0.44 ,
-0.69 , -0.04 , -0.28 , -0.48 , -0.01 , 0.00 , -0.03 ,
-0.10 , -0.25 , -0.05 , 0.05 , -0.12 , -0.20 , -0.11 ,
-0.04 , -0.51 , -0.06 , 0.05 , -0.32 , -0.45 , -0.30 ,
-0.49 , -0.02 , -0.16 , -0.31 , -0.02 , -0.00 , -0.02
-0.63 , -0.02 , -0.25 , -0.43 , -0.00 , -0.02 , -0.02
-0.45 , -0.03 , -0.14 , -0.27 , 0.00 , -0.01 , -0.04 ,

exemplar = 0.05

currnt exemplars not = previous exemplar

0.05 not = 0.01

continue iteration to check if exemplar change nor not ... if not change ... break the iteration loop

next iteration
-0.09 , -0.86 , -0.16 , 0.09 , -0.59 , -0.77 , -0.55
-0.90 , -0.02 , -0.36 , -0.61 , 0.02 , 0.02 , -0.02 ,
-0.15 , -0.31 , -0.07 , 0.07 , -0.14 , -0.25 , -0.13
-0.10 , -0.67 , -0.12 , 0.10 , -0.44 , -0.60 , -0.41
-0.65 , -0.01 , -0.21 , -0.39 , 0.00 , 0.02 , -0.01 ,
-0.84 , -0.01 , -0.32 , -0.55 , 0.02 , 0.00 , -0.01 ,
-0.59 , -0.02 , -0.18 , -0.34 , 0.02 , 0.02 , -0.02 ,

exemplar = 0.10,0.00,0.00
currnt exemplars not = previous exemplar
0.10,0.00,0.00 not = 0.05

continue iteration :
after 48 iteration we get

-0.16 , -1.29 , -0.27 , 0.16 , -0.94 , -1.22 , -0.82 ,
-1.38 , 0.00 , -0.53 , -0.89 , -0.03 , -0.03 , -0.00 ,
-0.27 , -0.44 , -0.11 , 0.11 , -0.24 , -0.41 , -0.16 ,
-0.62 , -1.43 , -0.62 , 0.41 , -1.13 , -1.37 , -1.02 ,
-1.25 , -0.19 , -0.55 , -0.82 , -0.03 , -0.21 , -0.16 ,
-1.53 , -0.16 , -0.72 , -1.06 , -0.21 , -0.03 , -0.19 ,
-0.91 , 0.00 , -0.25 , -0.49 , -0.03 , -0.03 , -0.00 ,

exemplars : 0.00, 0.41

next iteration

-0.16 , -1.29 , -0.27 , 0.16 , -0.94 , -1.22 , -0.82 ,
-1.38 , 0.00 , -0.53 , -0.89 , -0.03 , -0.03 , -0.00 ,
-0.27 , -0.44 , -0.11 , 0.11 , -0.24 , -0.41 , -0.16 ,
-0.62 , -1.43 , -0.62 , 0.41 , -1.13 , -1.37 , -1.02 ,
-1.25 , -0.19 , -0.55 , -0.82 , -0.03 , -0.21 , -0.16 ,
-1.53 , -0.16 , -0.72 , -1.06 , -0.21 , -0.03 , -0.19 ,
-0.91 , 0.00 , -0.25 , -0.49 , -0.03 , -0.03 , -0.00 ,

exemplars: 0.00, 0.41
as you see after 49 iteration
exemplars wan’t change.................
algorithm terminate

exemplar 0.00 position = 1
exemplar 0.41 position = 3

clustering .............
1- get max(similarities[1] , similarities[3])
for each i
for each k
if similarities[i][1] > similarities[i][3] ... dataset(i) is belong to cluster(1)
if similarities[i][1] <similarities[i][3] ... dataset(i) is belong to cluster(3)

similarities
-0.08 ,  -0.56 ,  -0.08 ,  -0.01 ,  -0.40 ,  -0.52 ,  -0.36 ,
-0.56 ,  -0.08 ,  -0.22 ,  -0.42 ,  -0.01 ,  -0.00 ,  -0.02 ,
-0.08 ,  -0.22 ,  -0.08 ,  -0.03 ,  -0.12 ,  -0.19 ,  -0.10 ,
-0.01 ,  -0.42 ,  -0.03 ,  -0.08 ,  -0.28 ,  -0.38 ,  -0.25 ,
-0.40 ,  -0.01 ,  -0.12 ,  -0.28 ,  -0.08 ,  -0.01 ,  -0.00 ,
-0.52 ,  -0.00 ,  -0.19 ,  -0.38 ,  -0.01 ,  -0.08 ,  -0.01 ,
-0.36 ,  -0.02 ,  -0.10 ,  -0.25 ,  -0.00 ,  -0.01 ,  -0.08

for i=0
-0.56 < -0.01
dataset(0) -> cluster 3

for i=1
-0.08 > -0.42
dataset(1) -> cluster 1

for i=2
-0.22 < -0.03
dataset(2) -> cluster 3

for i=3
-0.41 < -0.08
dataset(3) -> cluster 3

for i=4
-0.01> -0.28
dataset(4) -> cluster 1

for i=5
-0.00> -0.38
dataset(5) -> cluster 1

for i=6
-0.02> -0.25
dataset(6) -> cluster 1

cluster(1) = 0.15, 0.27, 0.18, 0.30
cluster(3) = 0.90, 0.62, 0.8

lets consider another exemple this time we work on 2-D ...points(x,y)

len: 25
(0,0) (0,2) (0,4) (0,6) (1,1)

(1,3) (1,5) (2,0) (2,2) (2,4)

(2,6) (3,1) (3,3) (3,5) (4,0)

(4,2) (4,4) (4,6) (5,1) (5,3)

(5,5) (6,0) (6,2) (6,4) (6,6)

similarities after setting diagonal:

len = 625

-16 -4 -16 -36 -2 -10 -26 -4 -8 -20 -40 -10 -18 -34 -16 -20 -32 -52 -26 -34 -50 -36 -40 -52 -72

-4 -16 -4 -16 -2 -2 -10 -8 -4 -8 -20 -10 -10 -18 -20 -16 -20 -32 -26 -26 -34 -40 -36 -40 -52

-16 -4 -16 -4 -10 -2 -2 -20 -8 -4 -8 -18 -10 -10 -32 -20 -16 -20 -34 -26 -26 -52 -40 -36 -40

-36 -16 -4 -16 -26 -10 -2 -40 -20 -8 -4 -34 -18 -10 -52 -32 -20 -16 -50 -34 -26 -72 -52 -40 -36

-2 -2 -10 -26 -16 -4 -16 -2 -2 -10 -26 -4 -8 -20 -10 -10 -18 -34 -16 -20 -32 -26 -26 -34 -50

-10 -2 -2 -10 -4 -16 -4 -10 -2 -2 -10 -8 -4 -8 -18 -10 -10 -18 -20 -16 -20 -34 -26 -26 -34

-26 -10 -2 -2 -16 -4 -16 -26 -10 -2 -2 -20 -8 -4 -34 -18 -10 -10 -32 -20 -16 -50 -34 -26 -26

-4 -8 -20 -40 -2 -10 -26 -16 -4 -16 -36 -2 -10 -26 -4 -8 -20 -40 -10 -18 -34 -16 -20 -32 -52

-8 -4 -8 -20 -2 -2 -10 -4 -16 -4 -16 -2 -2 -10 -8 -4 -8 -20 -10 -10 -18 -20 -16 -20 -32

-20 -8 -4 -8 -10 -2 -2 -16 -4 -16 -4 -10 -2 -2 -20 -8 -4 -8 -18 -10 -10 -32 -20 -16 -20

-40 -20 -8 -4 -26 -10 -2 -36 -16 -4 -16 -26 -10 -2 -40 -20 -8 -4 -34 -18 -10 -52 -32 -20 -16

-10 -10 -18 -34 -4 -8 -20 -2 -2 -10 -26 -16 -4 -16 -2 -2 -10 -26 -4 -8 -20 -10 -10 -18 -34

-18 -10 -10 -18 -8 -4 -8 -10 -2 -2 -10 -4 -16 -4 -10 -2 -2 -10 -8 -4 -8 -18 -10 -10 -18

-34 -18 -10 -10 -20 -8 -4 -26 -10 -2 -2 -16 -4 -16 -26 -10 -2 -2 -20 -8 -4 -34 -18 -10 -10

-16 -20 -32 -52 -10 -18 -34 -4 -8 -20 -40 -2 -10 -26 -16 -4 -16 -36 -2 -10 -26 -4 -8 -20 -40

-20 -16 -20 -32 -10 -10 -18 -8 -4 -8 -20 -2 -2 -10 -4 -16 -4 -16 -2 -2 -10 -8 -4 -8 -20

-32 -20 -16 -20 -18 -10 -10 -20 -8 -4 -8 -10 -2 -2 -16 -4 -16 -4 -10 -2 -2 -20 -8 -4 -8

-52 -32 -20 -16 -34 -18 -10 -40 -20 -8 -4 -26 -10 -2 -36 -16 -4 -16 -26 -10 -2 -40 -20 -8 -4

-26 -26 -34 -50 -16 -20 -32 -10 -10 -18 -34 -4 -8 -20 -2 -2 -10 -26 -16 -4 -16 -2 -2 -10 -26

-34 -26 -26 -34 -20 -16 -20 -18 -10 -10 -18 -8 -4 -8 -10 -2 -2 -10 -4 -16 -4 -10 -2 -2 -10

-50 -34 -26 -26 -32 -20 -16 -34 -18 -10 -10 -20 -8 -4 -26 -10 -2 -2 -16 -4 -16 -26 -10 -2 -2

-36 -40 -52 -72 -26 -34 -50 -16 -20 -32 -52 -10 -18 -34 -4 -8 -20 -40 -2 -10 -26 -16 -4 -16 -36

-40 -36 -40 -52 -26 -26 -34 -20 -16 -20 -32 -10 -10 -18 -8 -4 -8 -20 -2 -2 -10 -4 -16 -4 -16

-52 -40 -36 -40 -34 -26 -26 -32 -20 -16 -20 -18 -10 -10 -20 -8 -4 -8 -10 -2 -2 -16 -4 -16 -4

-72 -52 -40 -36 -50 -34 -26 -52 -32 -20 -16 -34 -18 -10 -40 -20 -8 -4 -26 -10 -2 -36 -16 -4 -16

As you see... a large matrix has been generated with len=625 where
25 * 25 = 625

“assume that you input 1k set of data... the simlarities size = 1k * 1k = 1mb

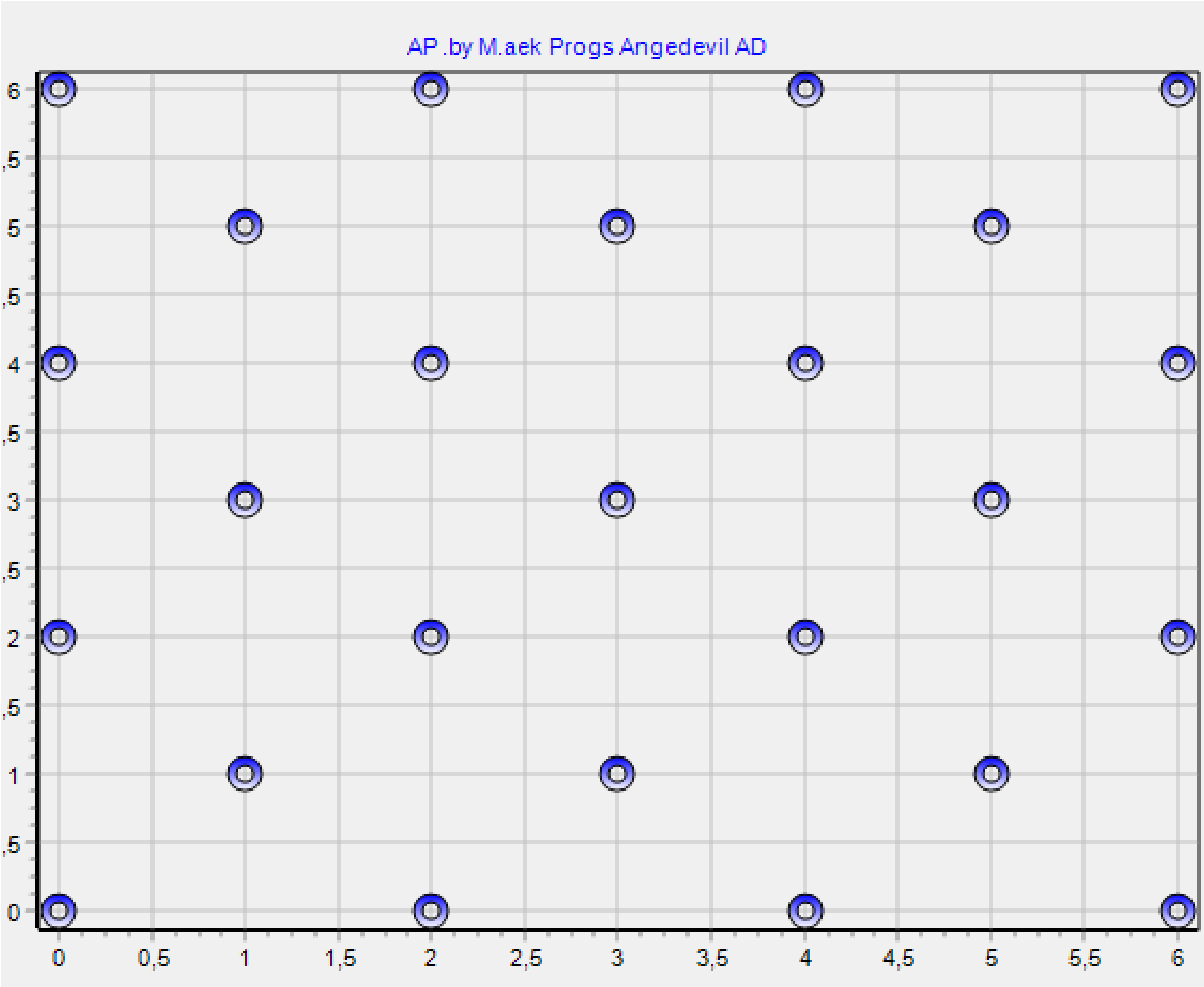
1mb set of data give you 1TB....1mb *1mb = 1000000*1000000 = 1000000000000

Space complexity is given by: $O(N^2)$

algorithm therminate after 27 iteration .. exemplar table:

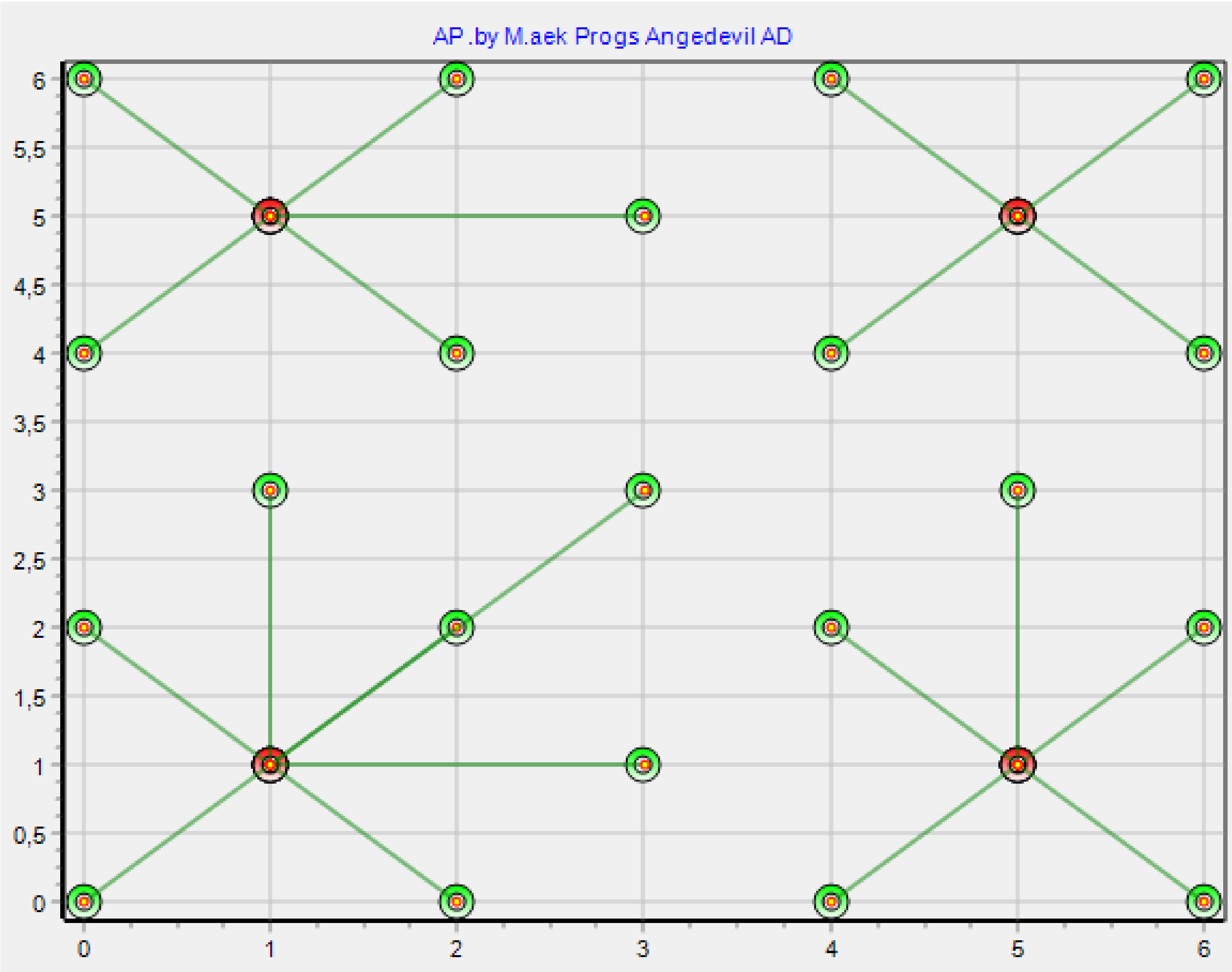
4,4,6,6,4, 4,6,4,4,6,
6,4,4,6,18,
18,20,20,18,18,
20,18,18,20,20

Time Complexity ....
$O(N^2 i)$



AP .by M.aek Progs Angedevil AD
6
,5
5
,5
4
,5
3
,5
2
,5
1
,5
0
0
0,5
1
1,5
2
2,5
3
3,5
4
4,5
5
5,5
6

dataset after affinity propagation




red point i = exemplars

affinity propagation doesn’t need to dertmine the number of cluster.
affinity propagation is better than kmeans & k medoids on error minimizing and outliers

## application:

-The inventors of affinity propagation showed it is better for certain computer vision and computational biology tasks
-Another recent application was in economics, when the affinity propagation was used to find some temporal patterns in the output multipliers of the US economy between 1997 and 2017
- ........and many other advantages